Intelligent Navigational Strategies for Multiple Wheeled Mobile Robots using Artificial Hybrid Methodologies

Bhumeshwar Kunjilal Patle

Department of Mechanical Engineering
National Institute of Technology Rourkela

Intelligent Navigational Strategies for Multiple Wheeled Mobile Robots using Artificial Hybrid Methodologies

Dissertation submitted to the

National Institute of Technology Rourkela

in partial fulfillment of the requirements

of the degree of

Doctor of Philosophy

in

Mechanical Engineering

by

Bhumeshwar Kunjilal Patle

(Roll Number: 511ME814)

under the supervision of

Prof. Dayal Ramakrushana Parhi

and

Prof. A. Jagadeesh

December, 2016

Department of Mechanical Engineering

National Institute of Technology Rourkela

Department of Mechanical Engineering
National Institute of Technology Rourkela

Dec 07, 2016

Certificate of Examination

Roll Number: 511ME814
Name: Bhumeshwar Kunjilal Patle
Title of Dissertation: Intelligent Navigational Strategies for Multiple Wheeled Mobile Robots Using Artificial Hybrid Methodologies

We the below signed, after checking the dissertation mentioned above and the official record books of the student, hereby state our approval of the dissertation submitted in partial fulfillment of the requirement of the degree of Doctor of Philosophy in Mechanical Engineering at National Institute of Technology, Rourkela. We are satisfied with the volume, quality, correctness, and originality of the work.

A. Jagadeesh Co-supervisor	Dayal R. Parhi Principal Supervisor
Susmita Das Member (DSC)	S. Murugan Member (DSC)
Hara Prasad Roy Member (DSC)	Rajeev Srivastava Examiner
S. K. Sahoo Chairman (DSC)	S. S. Mahapatra Head of the Department

Department of Mechanical Engineering
National Institute of Technology Rourkela

Dayal R. Parhi
Professor

A. Jagadeesh
Professor

Dec 07, 2016

Supervisor's Certificate

This is to certify that the work presented in this dissertation entitled *"Intelligent Navigational Strategies for Multiple Wheeled Mobile Robots using Artificial Hybrid Methodologies"* by *"Bhumeshwar Kunjilal Patle"*, Roll Number: 511ME814, is a record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of *Doctor of philosophy* in *Mechanical Engineering*. Neither this dissertation nor any part of it has been submitted for any degree or diploma to any institute or university in India or abroad.

A. Jagadeesh	Dayal R. Parhi
Co-supervisor	Principal Supervisor

To my Parents,
with all my love

Declaration of Originality

I, Bhumeshwar Kunjilal Patle, Roll Number: 511ME814 hereby declare that this dissertation entitled *"Intelligent Navigational Strategies for Multiple Wheeled Mobile Robots using Artificial Hybrid Methodologies"* represents my original work carried out as a doctoral student of NIT Rourkela and, to the best of my knowledge, it contains no material previously published or written by another person, nor any material presented for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the section "Bibliography". I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of my non-compliance detected in the future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

Dec 07, 2016

NIT Rourkela

Bhumeshwar Kunjilal Patle

Acknowledgment

My first thank is to the Almighty God, without whose blessings, I wouldn't have been writing this “acknowledgments". I would like to extend my heartfelt indebtedness and gratitude to Prof. Dayal R. Parhi and Prof. A. Jagadeesh for his kindness in providing me an opportunity to work under his supervision and guidance. During this period, without his endless efforts, immense knowledge, deep patience, invaluable guidance and answers to my numerous questions, this research would have never been possible. I am especially obliged to him for teaching me both research and writing skills, which have been proven beneficial for my current research and future career. He showed me different ways to approach a research problem and the need to be persistent to accomplish any goal. It has been a great honor and pleasure for me to do research under the supervision of Dr. Dayal R. Parhi and Dr. A. Jagadeesh. I am thankful to Prof. Animesh Biswas, Director of National Institute of Technology, for giving me an opportunity to be a part of this institute of national importance and to work under the supervision of Prof. Dayal R. Parhi. I am sincerely obliged to Prof. S. S. Mahapatra, Head of the Department, Department of Mechanical Engineering, for providing me all official and laboratory facilities during the research period. His incessant encouragement towards research work has inspired me a lot. I also thankful to management of RSR RCET Bhilai specially to Mr. Sanjay Rungta, Prof. A. Jagadeesh, Mr. Saket Rungta, Prof. P. S. Bokare and Prof. S. S. Das for sponsoring me to carryout PhD work. I express my gratitude to Prof. S. K. Sahoo, Chairman DSC, Prof. S. Murugan, Prof. Susmita Das and Prof. Hara Prasad Roy for their indebted help and valuable suggestions for the accomplishment of the dissertation. I thank all the members of the Department of Mechanical Engineering, and the Institute, who helped me in various ways towards the completion of my work.

I would like to thank all my friends and lab-mates Maheswar, Anish, Prases, Alok, Dr. Sunil Kumar Kashyap for their encouragement and understanding. Their support and lots of lovely memories with them can never be captured in words. Finally, I thank my parents, beloved wife Leena, my baby Reyansh and the entire family members for their unlimited support and strength.

Dec 07, 2016
NIT Rourkela

Bhumeshwar Kunjilal Patle
Roll Number: 511ME814

Abstract

At present time, the application of mobile robot is commonly seen in every fields of science and engineering. The application is not only limited to industries but also in the household, medical, defense, transportation, space and much more. They can perform all kind of tasks which human being cannot do efficiently and accurately such as working in hazardous and highly risk condition, space research etc. Hence, the autonomous navigation of mobile robot is the highly discussed topic of today in an uncertain environment. The present work concentrates on the implementation of the Artificial Intelligence approaches for the mobile robot navigation in an uncertain environment. The obstacle avoidance and optimal path planning is the key issue in autonomous navigation, which is solved in the present work by using artificial intelligent approaches. The methods use for the navigational accuracy and efficiency are Firefly Algorithm (FA), Probability-Fuzzy Logic (PFL), Matrix based Genetic Algorithm (MGA) and Hybrid controller (FA-PFL, FA-MGA, FA-PFL-MGA).The proposed work provides an effective navigation of single and multiple mobile robots in both static and dynamic environment. The simulational analysis is carried over the Matlab software and then it is implemented on a mobile robot for real-time navigation analysis. During the analysis of the proposed controller, it has been noticed that the Firefly Algorithm performs well as compared to fuzzy and genetic algorithm controller. It also plays an important role in building the successful Hybrid approaches such as FA-PFL, FA-MGA, FA-PFL-MGA. The proposed hybrid methodology perform well over the individual controller especially for path optimality and navigational time. The developed controller also proves to be efficient when they are compared with other navigational controller such as Neural Network, Ant Colony Algorithm, Particle Swarm Optimization, Neuro-Fuzzy etc.

Keywords: Firefly Algorithm, Genetic Algorithm, Fuzzy-Logic, Mobile Robot Navigation, Hybrid Controller

Contents

List of Figures

List of Tables

Nomenclatures

V_R	Right Wheel Linear Velocity
V_L	Left Wheel Linear Velocity
ω_R	Angular Velocity of Right Wheel
ω_L	Angular Velocity of Left Wheel
θ	Steering Angle (Turning Angle)
C	Center of Mass of a Mobile Robot
R	Radius of Wheel
V	Centre Linear Velocity of the Robot
ω	Centre Angular (Rotational) Velocity of Left Wheel
L	Track Width of the Robot
m	Total Mass of the Mobile Robot
I	Moment of Inertia of the Robot
τ_R	Right Wheel (Motor) Torques
τ_L	Left Wheel (Motor) Torques
d_f	Forward Obstacle Distance
d_l	Left Forward Obstacle Distance
d_r	Right Forward Obstacle Distance
m_r	Right Motor Velocity
m_l	Left Motor Velocity
F.O.D.	Front Obstacle Distance
L.O.D.	Left Obstacle Distance
R.O.D.	Right Obstacle Distance
H.A.	Heading Angle

RV	Right Wheel Velocity
LV	Left Wheel Velocity
FLA	Fuzzy logic architecture
SA	Simulated Annealing Algorithm
GA	Genetic Algorithm
PSO	Particle Swarm Optimization Algorithm
ACO	Ant Colony Optimization Algorithm
FA	Firefly Algorithm
PFL	Probability based Fuzzy-Logic
MGA	Matrix based Genetic Algorithm
CS	Cuckoo Search Algorithm
BFO	Bacterial Forging Optimization
ABC	Artificial Bee Colony
IWO	Invasive Weed Optimization
SFLA	Shuffeled Frog Leaping Algorithm
BA	Bat Algorithm
MRN	Mobile Robot Navigation
WMRN	Wheeled Mobile Robot Navigation
β	The angle of the wheel plane relative to the chassis

Note: - The symbols and abbreviations other than above have been explained in the text.

Chapter 1

Introduction

The proposed work in the field of mobile robot navigation addresses the potential of Artificial Intelligent (AI) methods for design and development of the path planning and control strategies for mobile robotics. The chapters included in the thesis have been classified into four main sections. The first section of the chapter deals with the background and inspiration behind the proposed research work, whereas the second section discusses objective of the work and its scope in the field of engineering and science. The originality of the work is presented in the third section. The outline of all chapters of the thesis work is concluded in the fourth section.

1.1 Background and Inspiration

At present, in all the fields of science and engineering from industry to household, medical to military are commonly using the robots. Its success and desirable outcome make it suitable to accomplish the needed task, and so it is highly researched topic of today. Industrial and technical applications of mobile robots are continuously gaining in importance, in particular under considerations of reliability (uninterrupted and reliable execution of monotonous tasks such as surveillance), accessibility (inspection of sites that are inaccessible to humans, e.g. tight spaces, hazardous environments or remote sites) or cost (transportation systems based on autonomous mobile robots can be cheaper than standard track-bound systems). The present mobile robots can be used for surveillance, inspection, entertainment and transportation tasks. The main application of mobile robot is seen in the dangerous field such as mining industry, nuclear industry, space research and landmine detection in the military operation where the human interaction may cause accidents. To achieve safe path and a successful navigation in such dangerous field is a challenging task for any automobile robot. So, attention on path planning strategy to make automobile robot navigation from initial position to destination by avoiding the obstacle is a fundamental need. Additionally, to minimize required time of navigation, energy consumption and communication delay, the safely organized path is required which should be optimal regarding path length.

The autonomous mobile robot is an artificially intelligent machine which is capable of understanding the environmental condition (position of obstacle and goal), able to do self-path planning (by avoiding the static and dynamic obstacle) and should be capable to quickly respond to any environmental condition without any human effort. Practical path planning in an uncertain environment is still a major problem in mobile robot navigation. At present Scenario, day by day real time implementation of automobile robot is continuously growing, and therefore, the automobile robot with efficient obstacle avoidance mechanism is need of today. The autonomous navigation of mobile robot is a complicated process not only about the determination of its position in its frame of reference but also about to plan towards the goal. The method of navigation consists of four main stages (shown in Figure 1.1) and are as follows,

- Perception
- Localization / Mapping
- Cognition / Planning
- Motion control

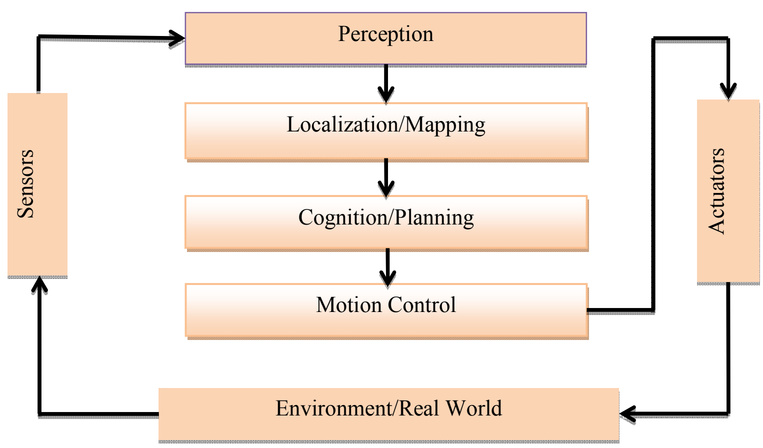


Figure 1.1: Sequential task of navigation process

With the help of sensor, the prior information of the environment is collected and this information is used to build the map of the surrounding (perception). The information obtained from the sensor is used to determine the position of the robot in the robot environment (localization). After localization, the robot must plan the path from the initial position to target position (Cognition / local path planning) and control the motion of the robot actuators (for motion control). By following the above basic steps of navigation, the desired path planning strategy for mobile robot navigation is formulated, which is capable of finding an optimal collision-free path from the initial position of the robot to a goal position in the uncertain environment. In mobile robot navigation, tracks, wheels and legs are used for the locomotion purpose. From the last decades, the mobile robot equipped with the wheel mechanism is popularly seen in industry to the household application for operation, transportation, and inspection. The research work presented in the thesis follows the wheeled mechanism for navigation in the uncertain environment.

The mobile robot navigation is not a big issue when the environment is without obstacle, but when the environment is filled with various static and dynamic obstacles, then it becomes the topic of research for optimization. Many researchers have provided the different approaches to solve the problem of navigation when the environment is known and unknown. The path planning approaches are broadly categorized as follows,

- Global path planning (Offline path planning) approach.
- Local path planning (Online path planning) approach.

In global path planning approaches, the initial information about the environment i.e. the position, shape, size of the obstacle are required for path planning whereas, in local path planning methods, no preliminary data of environment is necessary. On comparison, the local path planning approaches popularly used over global path planning approaches concerning low computational cost, real time implementation and capability to handle the uncertainty present in the environment. The traditional global path planning approaches such as Cell decomposition, Roadmap, Subgoal network, Artificial potential field and Voronoi diagram are not suitable for on-line implementation. Therefore, artificial intelligence approaches (for local path planning) such as Genetic Algorithm (GA), Neural Network (NN), Fuzzy Logic (FL), Bacteria Forging Optimization algorithm (BFOA), Ant colony algorithm (ACO), Cuckoo search algorithm (CSA), Particle Swarm Optimization (PSO), Bee algorithm (BA), Firefly Algorithm (FA), Simulated Annealing (SA), and combination of the above (Hybrid algorithm) have been used for online implementation of mobile robot navigation problem.

The work in thesis dedicates to design and development of artificial intelligent navigational strategies for multiple wheeled mobile robots in an uncertain environment by using the hybrid algorithm. To achieve the said goal, the Matrix based Genetic Algorithm (MGA), Probability-Fuzzy Logic (PFL), Firefly Algorithm (FA) and Hybrid Algorithms (such as FA-MGA, FA-PFL, FA-PFL-MGA) are studied to build real-time navigational path planner for single and multiple mobile robots. The work consists of the design and development of an intelligent controller to avoid the static and dynamic obstacle in minimum travel time. The analyzed advantage of the work can be easily implemented to design and development of the hybrid methodologies in minimum infrastructure. The useful hybrid controllers are designed and developed by hybridization of the intelligent controllers. These hybrid controllers are tested for different situations and are implemented for the computer based simulation to check feasibility over the uncertain environment. At last, the real-time navigation is demonstrated by developed controller on the real robot to validate the effectiveness at the proposed methodologies. The developed hybrid controller using probability-fuzzy logic, matrix based genetic algorithm and firefly algorithm are observed more advantageous when compared with a single controller in terms of path length and time taken during navigation.

1.2 Aims and Objectives of Proposed Research Work

The principle goal of the current investigation is to design the artificial intelligent hybrid controller for effective path planning in the presence of a static and dynamic obstacles in the uncertain environment. The navigational approach is not only developed for the single mobile robot but also for multiple mobile robots. In this proposed work, Matrix based Genetic Algorithm (MGA), Probability-Fuzzy-Logic (PFL), Firefly Algorithm (FA) and the hybrid algorithm (FA-MGA, FA-PFL, FA-PFL-MGA) have been analyzed and employed to solve the mobile robot navigation problem. Specifically, the work wishes to observe the suitability of the hybrid controllers for effective path planning for single and multiple robots in the presence obstacles.

The principle objectives of the proposed work presented in the thesis are as follows:

- To carry out the kinematic analysis of wheeled mobile robot.
- To design and develop the matrix based genetic algorithm for developing a effective navigational strategy for mobile robot navigation problem.
- To generate the active rule mechanism by using probability-fuzzy logic for mobile robot navigation problem.
- To build up firefly algorithm based navigational path planning controller for mobile robot navigation.

- To develop the hybrid navigational controller based on firefly algorithm and matrix based genetic algorithm i.e FA-MGA.
- To develop the hybrid navigational controller for robot based on firefly algorithm and probability-fuzzy logic i.e FA-PFL.
- To develop the hybrid navigational controller for robot based on firefly algorithm, probability-fuzzy logic and matrix based genetic algorithm i.e. FA-PFL-MGA.
- To perform the simulation and experimental analysis of proposed methodologies for validation purpose.

In addition to said objectives the robot must have the following ability:

The robot must understand the data given by the sensors and able to understand the environment.

- It must be self-moving in its environment without slipping.
- It should have the proper obstacles detection and obstacles avoidance mechanism.
- It should not cause any damage to the environment.
- It must be intelligent to update itself from the self-learning ability for efficient searching.

Some extraordinary behaviors are given below for useful mobile robot navigation to achieve the above goals:

- **Goal seeking behavior:** With this behavior robot must search the target continuously till it reaches.
- **Obstacle avoidance behavior:** When the robots path consists of the obstacle then this behavior helps the robot to make safe distance with the obstacles and performs the obstacle avoidance task.
- **Wall following behavior:** Due to this behavior the robot can come out from the trap like situation. This mechanism helps the robot to follow the walls of the obstacle during navigation.

1.3 Novelty of the Proposed Research Work

The proposed research work in the thesis gives the novel hybrid controller for effective path planning in the uncertain environment in the presence of static and dynamic obstacles for multiple wheeled mobile robots. The three popular approaches such as genetic algorithm, fuzzy logic, and firefly algorithm are hybridized to get the benefit over the other approaches. As per the knowledge of the author, the newly discovered firefly algorithm is not yet hybridized with fuzzy logic and genetic algorithm for path planning problems of multiple wheeled mobile robots in a static and dynamic environment. The matrix based genetic algorithm and use of probability along with the fuzzy logic is the additional finding of the proposed research work.

1.4 Outline of the Thesis

The thesis is categorized in following sections as chapter wises:

Chapter-1 gives the brief introduction of mobile robot navigation, idea behind the proposed research work and objective.

Chapter-2 displays the detailed literature survey on different mobile robot navigational approaches.

Chapter-3 focuses the kinematic analysis of the wheeled mobile robot.

Chapter-4 presents the application of the matrix based genetic algorithm for the mobile robot path planning problem by finding the fittest chromosome among the population as the new position of the robot and maintains the diversity in population to get an optimal solution.

Chapter-5 deals with the use of the fuzzy logic technique along with the probability for path planning of mobile robot in the uncertain environment by generating the active rules.

Chapter-6 provides the application of firefly algorithm for mobile robot navigation. The fitness function is derived using biological mechanism of fireflies, for safe path planning and obstacle avoidance in a static and dynamic environment.

Chapter-7 gives the hybrid controller based on the matrix based genetic algorithm, probability-fuzzy logic and firefly algorithm. The designed controller performs better over the individual probability-fuzzy logic, matrix based genetic algorithm, and firefly algorithm.

Chapter-8 discusses the comprehensive final review of all discussed approaches on the basis of applicability.

Chapter-9 concludes the research work carried in this thesis and gives the positive approach towards the future application and research.

Chapter 2

Literature Review

This chapter focuses the highlights on the various research methodologies developed in the field of mobile robot navigation till now in context to the current research. The step by step investigations of classical and reactive approaches are made here to understand the development of path planning strategies in various environmental conditions. At the end of the chapter the summary of the literature is provided and effort has been given to find an appropriate gap or methodologies weakness in the existing study area to solve the research problem.

2.1 Introduction

Autonomous mobile robot path planning is the task of getting the appropriate movement in the uncertain environment without any human interference. The appropriate movement initiates the robot to attain a goal and during this, it has to detect and avoid collision with obstacles. The mobile robot and its environment must be quantified during path planning problem. The mobile robot model has its dimensions, differential equation, kinematics, control parameter over robot movement. Model of the environment has the position of robot and obstacle, map representation. For any mobile robot, self-localization, path planning, map building and obstacle avoidance are the requirements of navigation. Robot localization denotes robot ability to establish its own position and orientation within the frame of reference. Path planning is the extension of the localization in which it requires the determination of the robots current position and a position of a goal location, both within the frame of reference. Map building can be in the shape of a metric map or any notation describing the location in the robot frame of reference. In obstacle avoidance the robot responds to the environment by sensing obstacles. Global navigation, local navigation and personal navigation are the three different aspects of the mobile robot navigation. The ability to determine one's position in absolute or map-referenced terms and to move towards desired destination point is the global navigation. Local navigation is the ability to determine one's position relative to stationary or moving object in the environment and not to collide with them as one move. Being aware of the positioning of the various parts that make up one in relation to each other and handling the objects is the personal navigation.

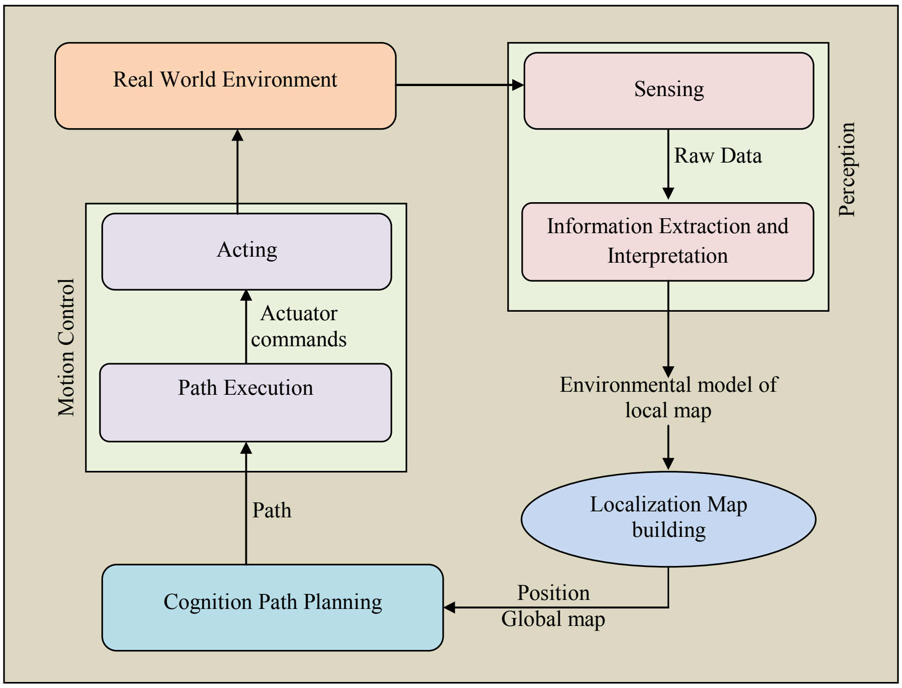


Figure 2.1: Flow diagram for mobile robot navigation (Horizontal decomposition)

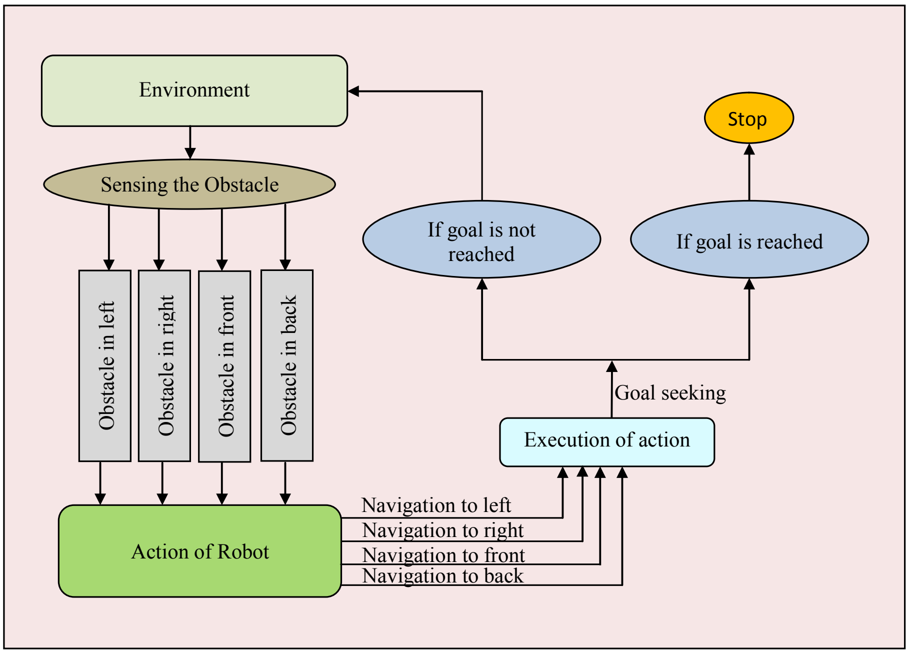


Figure 2.2: Flow diagram for mobile robot navigation (Vertical decomposition)

To solve the difficulties of the path planning problem, conventional and reactive approaches have been considered for the study. The most of conventional approaches are deterministic and it fails when there is the discontinuity in an objective function. However, reactive approaches have the ability to search space on the global platform to give up the diverse solution and to look for the feasible solution in the local region. In navigational problem, the collision free paths are constructed by the path planning algorithms, and robot moves along the constructed paths to reach the target. The path planning system for the mobile robots is decomposed into a series of functional units, as shown in Figure 2.1 by continuous vertical slices. After deciding the computational requirements for a robot, the path planning system is decomposed into a series of horizontal functional units to achieve the desire task behavior required for the robot (Figure 2.2). After, surveying many research articles in the robot path planning field, many existing research works for each technique is identified and categorized.

2.2 Kinematic Analysis of Wheeled Mobile Robot

2.2.1 Introduction

Kinematics is the most fundamental study associated with the operation of the mechanical system. In mobile robotics, kinematics related to the mechanical behavior of the robot while neglecting the effect of the forces acting on it. While designing a mobile robot for a particular application one has to consider the mechanical behavior of the system. The next step is to develop control software to attain thorough command over the hardware of the mobile robot.

2.2.2 Wheeled Locomotion for Mobile Robot

The application of mobile robot is increasing day by day in the field of medical sciences, the military operation of search and rescue, household work to industrial process, entertainment to the creation, space research, mining operation and much more. To perform this efficiently the robot requires appropriate locomotion mechanism. The locomotion mechanism equipped with legs is having some shortfalls that they lose energy and suffers from the high mechanical complexity and it requires a high degree of freedom. For effective autonomous mobile robot navigation, the wheeled locomotion mechanism [1-4] is popularly used. In most of industrial and household purposes, the mobile robots with motorized wheels are practiced for navigation on the flat and uneven ground. The wheeled mechanism design is simpler, easy to build, inexpensive and easy to control the movement. The autonomous mobile robot may have many wheels, but for satisfactory balance three wheels are sufficient [5-6]. However, the additional wheel can be used for balancing purpose when the ground is uneven. Apart from the balancing of the robot, the problems like control, stability and maneuverability were the great challenges to control velocity over the wheeled robot. To monitor the motion and according to the application the standard wheel, castor wheel, Swedish wheel and ball or spherical wheel is used due to having the significant effect on kinematics [7-8]. The Standard and castor wheel have significant influence of on robot locomotion as standard wheels give smooth motion without any effect whereas the castor wheels exert the force on the robot chassis during steering [9]. On the other hand in [10], the Swedish wheel functions like a normal wheel but it has some constrained in another direction. The wheels like spherical are called omnidirectional wheel as they have no constrained for the direction of motion as it can spin along any direction [11]. While selecting the wheel for the robot, the suspension system plays a significant role in any kind of terrain to maintain proper contact with the ground. So, in many robots, soft rubber is used to create an initial suspension for uneven terrain. Like proper wheel selection, the study of the wheel geometry which consists maneuverability, controllability and stability also key parameters while controlling kinematics of the robot [12]. The most of the automobile works in the highly uniform environment, however, the automobile robot has designed for numerous situations. In the case of the automobile, the maneuverability, controllability and stability remain maximum as they have same wheel configuration for their standard environment, but there is no single wheel configuration for automobile robot to achieve maximum maneuverability, controllability and stability in a variety of environment [13]. For stability point of view, the robot requires minimum two wheels. To get static stability in two wheel drive robot, the center of mass must act below the wheel axle. Alexander et al. [14] have correlated the robot motion, types of wheel drive and the connection between bodies for robot stability. They used the simple wheels for locomotion with the implementation of forward and reverse kinematics. To control the robot from skidding on the plane ground the Tsuchiya et al. [15] presented the new strategy whereas Mester [16] introduced the "Feed forward compensator" for modeling and controlling robot motion for uneven terrain. The analysis is carried out on two wheeled drive robot with independent angular velocities of the wheels. The kinematic analysis of three-wheeled (omnidirectional) mobile robot by geometric strategies is presented in [17]. To provide omnidirectional motion, the new wheel mechanism is designed and developed for the holonomic mobile platform by using three self-steered wheels. The problem of motion control along with kinematics and singularity analysis for Swedish wheel is presented by the Giovanni [18]. Wada et al. [19]

have developed improved wheel mechanism for holonomic and omnidirectional robots. They used Synchro-caster wheel drive mechanism with self-governing decoupled gear train. To select the proper wheel for required operation the kinematic and dynamic analysis of wheels has been tested with consideration of skidding and sliding velocities [20]. While testing, method of augmented generalized coordinates has been used to carry out forward and inverse kinematic model. The same approach is also used by [21] to match the input vector and output vector of the mobile robot. To study the kinematics of mobile robots, the matrix coordinates transformation approach has proposed by [22]. The proposed approach gives satisfactory result when tested on a tricycle for forward velocity kinematics. Borenstein [23] have discovered compliant linkage mechanism for controlling and designing of the multi-degree of freedom mobile robots. The new device helps in minimizing the error and slipping. To improve the performance of the mobile robot, the variable length axle is presented in [24] over the rigid axle to minimize the slip. The artificial intelligence technique like fuzzy logic [25] and genetic algorithm [26] is used as the control strategy for the mobile robot. Teimoori et al. [27] have presented new guidance algorithm to drive wheeled robot toward the static and moving target based on the range only measurement. The proposed approach generates an equiangular spiral trajectory for locomotion. Zheng-Cai et al. [28] presented the point stabilization scheme for the wheeled mobile robot for uneven surfaces by using the fuzzy-genetic algorithm. The fuzzy logic used to control the speed and angular velocity where the genetic algorithm is used to optimize the control parameters. Eghtesad et al. [29] have presented the combined open/close loop method and feedback linearization approach for stabilizing the center of mass of the vehicle during the curvilinear motion. Mekkonnen et al. [30] have presented the position based visual servoing and image based visual servoing strategy which helps for steering towards the specific goal in the environment without requiring any prior information of the environment. Grand et al. [31] presented the analysis of the wheeled mobile locomotion on rough terrain by using the principles of the velocities to link the operational and joint parameter, the principle of virtual work to connect the contact forces, gravitational force and joint torques. The results show the efficient control over the posture of the robot in the static and dynamic environment. Chakraborty et al. [32] presented wheeled mobile robot navigation in uneven terrain without sleep by using torus wheel with a single point of contact. Kalinski et al. [33] introduced the optimal control strategy for the two-wheeled mobile robot based on energy performance and it is efficient for the problem of motion surveillance.

2.3 Navigation Technique used for Mobile Robot

Continuous research in the field of mobile robot navigation leads to the existence of effective navigational technique for controlling and guiding the robot for industrial and household purposes. Various researcher and scientist, from last few decades, have provided numerous studies on navigational approaches to find a suitable methodology for controlling the robots. The current research work made in thesis devoted to the development of efficient path planning for single and multiple mobile robots by using the intelligent hybrid approaches in the static and dynamic environment. The various methods employed for the navigation of mobile robot are broadly classified into two categories (classical and reactive approaches) as discussed below.

2.3.1 Classical Approaches

The many classical approaches are used to solve the navigational problem of the mobile robot. The reviews based on the classical methods are described below.

2.3.1.1 Cell Decomposition Approach

It is one of the popular approaches used for path planning in mobile robotics. Cell decomposition approach divides the region into the non-overlapping grids (cell) and uses the connectivity graphs for traversing from one cell to another cell in order to achieve the goal [34-36]. During the traversing, the pure cells (cell without obstacle) are considered to achieve the path planning from the initial position to target position. The corrupted cells (cells with the obstacle) present in the path are further divided into two new cells to get pure cell and this pure cell added to the sequence while getting the optimal path from the initial position to target position. In cell decomposition approach, the initial position and target position are represented by the starting and ending cells. The sequence of pure cells that joins these positions shows the required path [37].

Cell decomposition approach is divided into three parts

- Exact cell decomposition.
- Approximate cell decomposition.
- Adaptive cell decomposition

In the exact cell decomposition [38-39] shown in Figure 2.3, cells do not have specific shape and size, but it can be determined by the map of environment and shape and location of the obstacle within it. This method uses the regular grid in a various way. The first step in this type of cell decomposition is to decompose the free space, which is bounded both externally and internally by polygons, into trapezoidal and triangular cells by simply drawing parallel line segments from each vertex of each interior polygon in the configuration space to the exterior boundary. Then each cell is numbered and represented as a node in the connectivity graph. Nodes that are adjacent in the configuration space are linked in the connectivity graph. A path in this graph corresponds to a channel in free space, which is illustrated by the sequence of striped cells. This channel is then translated into a free path by connecting the initial configuration to the goal configuration through the midpoints of the intersections of the adjacent cells in the channel.

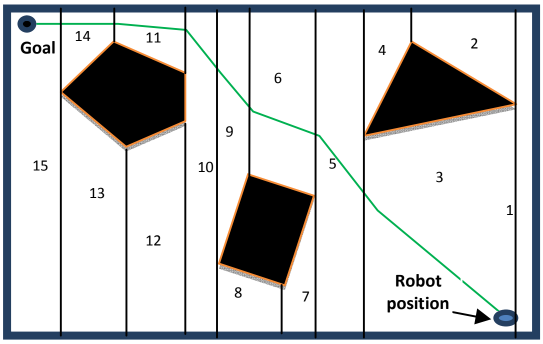


Figure 2.3: Exact cell decomposition

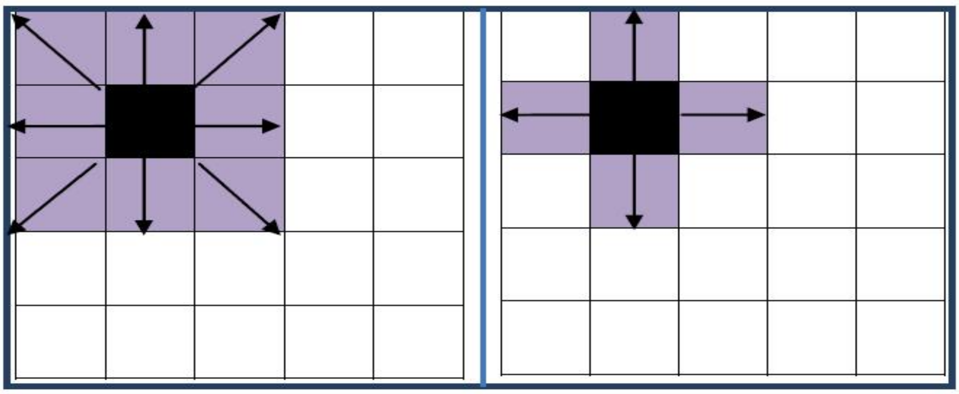


Figure 2.4: Approximate cell decomposition (8-connected and 4-connected grids)

In approximate cell decomposition [40-41], a regular grid is placed over the planning space and all the cells of the grid are predefined in shape and size so that it is easy to apply. This method of cell decomposition is called “approximate” because the boundaries of the physical objects do not need to coincide with the predefined cell boundaries. Any object placed in the grid area is considered as an obstacle else it is left as free space. To find a path, the center of each cell is taken as a node in the search graph. As shown in

Figure 2.4, these nodes can either be 4-connected or 8-connected representing whether or not the robot is considered to travel diagonally between them.

Adaptive cell decomposition understands the information present in free space and follows the basic concept of avoidance of the free space in regular cell decomposition. Samet [42] and Noborio [43] have proposed quadtree-adaptive decomposition. It started to divide the environment by large size cell but when the grid cell is partially occupied then in such condition it divides it into four equal subparts. These subparts are then subdivided again until each of the cells is either entirely full or empty. The resulting map has grid cells of varying size and concentration, but the cell boundaries coincide very closely with the obstacle boundaries as shown in Figure 2.5. When the robot acquires new data and updates its map based on new obstacles, then adaptive cell decomposition causes problems for dynamic environments. Hence, it is necessary for the entire data structure of the map to be completely restored.

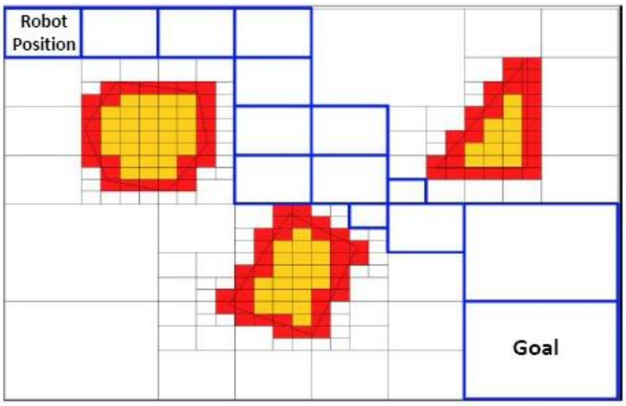


Figure 2.5: Adaptive cell decomposition

2.3.1.2 Roadmap Approach

Roadmap approach is also called as highway approach. Roadmap is the way to get from one place to another and it is the connection between free spaces is represented as a set of the one-dimensional curve [44]. After building the roadmaps, it used as a set of homogenous paths which the planer will search through to find optimal solution. The nodes in the graphs are usually waypoints that the robot needs to travel between for a successful journey. Therefore, roadmap approach is used to find shortest path from robot initial position to its target position. Voronoi and Visibility graphs are used to develop the roadmap. The visibility graph method connects the initial and the goal position with nodes from the map and searches for the path. Figure 2.6 represents the visibility graph where the dark area shows obstacle and dashed line shows the respective path from the initial position to goal position [45]. This method is also used for the environment with polygonal obstacles in which the vertices of the polygon is represented by the nodes and edges as a connector between the nodes [46].

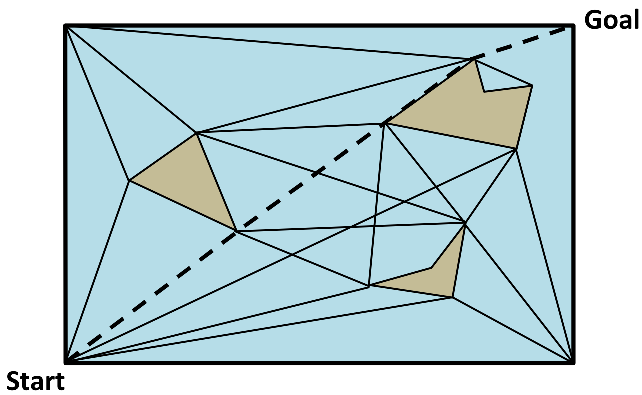


Figure 2.6: Visibility graph

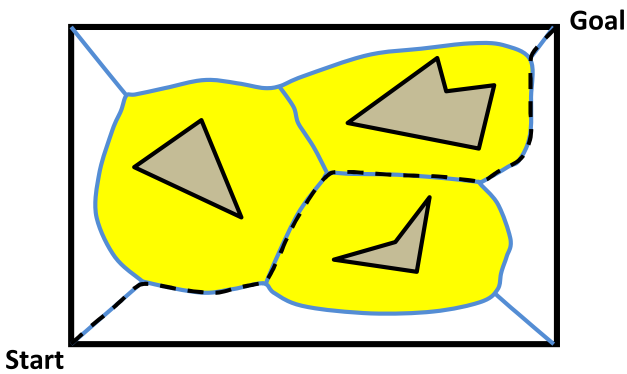


Figure 2.7: Voronoi diagram

The Voronoi diagram [47-49] is another roadmap algorithm used for the path planning of the robot. This method makes region into sub-region where all edges of the Figure are constructed using equidistant points from the adjacent two points on the obstacles boundaries. The Figure 2.7 represents the working of the Voronoi diagram.

The application of Voronoi diagram in the field of mobile robot navigation is presented by the [50-52] to keep robot away from the obstacle while moving in the environment. To improve the performance and to eliminate the drawback such as sharp turns and long loops in Voronoi diagram, some improvement is provided for effective path planning [53]. The hybrid approach is developed by combining the visibility graph, Voronoi diagram and potential field method [54] to get path optimality. Perhaps it has been noticed that the approach fails to get optimal path and execution process were complicated. To develop successful path planning by Voronoi diagram various strategies were implemented like skeleton maps by Yang et al. [55]. On the other hand, the combined approach of visibility graph and Voronoi diagram is presented by the Wein et al. [56] have provided the optimal root in planer environment. Kavraki et al. [57] presented the application of probability for Roadmap approach to understand and generate the solution to path planning. However, the approach is not efficient to get optimum path length. To obtain improvement in the process of finding the shortest path, Sanchez et al. [58] made a little variation in probabilistic roadmap approach by using the lazy-in-collision-detection method.

2.3.1.3 Artificial Potential Field Approach

Khatib [59] in 1986 presented the artificial potential field approach for mobile robot navigation. According to him, goal and obstacles act like charged surfaces and the total potential creates the imaginary force on the robot. This imaginary force attracts the robot towards the goal and keeps away from the obstacle as shown in Figure 2.8. The robot follows the negative gradient to avoid the collision caused by the obstacles and reach the target points. This method is used by various researchers for effective path planning and obstacle avoidance of mobile robot. The potential field method for mobile robot navigation is presented by the Garibotto et al. [60]. The new obstacle avoidance strategy in an unknown environment is discussed by the Kim et al. [61] by using potential field approach. They used a harmonic function to avoid a local minimum problem. The Borenstein et al. [62] have also presented the solution to the problem of the local minima condition. In this, they have considered the dynamic properties of robot navigation. The analysis of potential field method in the dynamic environment for obstacle avoidance is performed in [63]. The new improvisation in potential field method is done by using laws of electrostatic [64]. The implementation of the electrostatic helps to produce the potential function and to get the collision-free path in real time. The moving obstacle avoidance in real time is not so easy and hence Huang [65] developed the velocity controlling mechanism to understand the location and velocity of the obstacle while achieving the goal. To avoid local minima, the superior potential function and superior repulsive potential function is introduced by Shi et al. [66] to achieve global optima. Sfeir et al. [67] solve the observed problem in mobile robot navigation by potential field approaches such as oscillation and conflicts. They have presented improved version of the potential field algorithm to minimize the oscillation and conflicts when the goal is closer to the obstacle. They also provided the rotational force to produce a better path in the presence of the obstacle. Again, improved potential field approach is provided by Biswas et al. [68] to avoid the oscillations problem. The simultaneous comparison is made between the two methods i.e. traditional method and Levenberg-Marquardt method and it has been concluded that the solution obtained from the Levenberg-Marquardt is better as compared to traditional algorithm. The proposed approach minimizes the oscillation and produces the collision-free path. The problem of multiple mobile robot navigations is solved by applying potential field method in [69, 63]. The application of finite element method (FEM) along with potential field is discussed in [70]. The presented approach transforms the navigational problem into the electrostatic problem and then solved by the FEM. To test the applicability of the potential field approach Pradhan et al. [71] used ROBOPATH simulational tool. The multiple mobile robots are considered for various environmental condition and they observed better result in coordination strategy without collision.

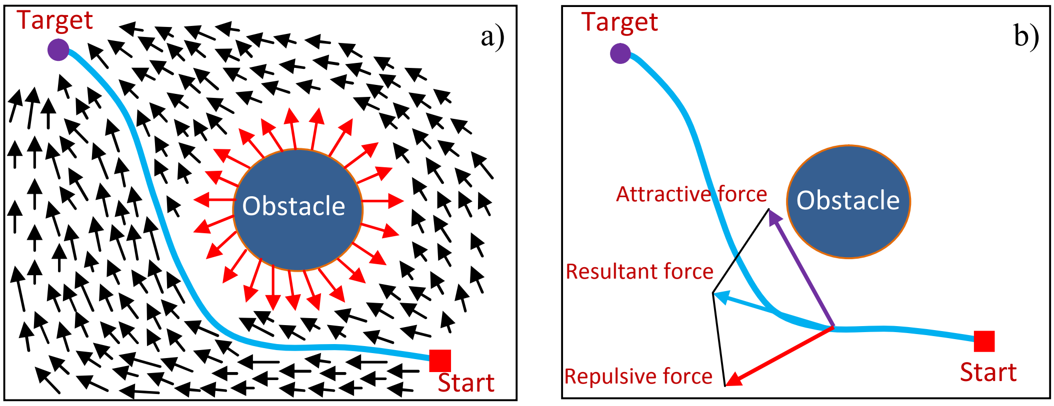


Figure 2.8: Mobile robot navigation by artificial potential field approach

2.3.2 Computational Intelligence Approach

Now days, a computational intelligent approach is accepted as most popular tool for mobile robot navigation when compared with the classical approach. It has great ability to handle the uncertainty present in the environment. There are various techniques developed for the navigation of the mobile robot in known and unknown environment. Parhi [72-73] have classified the navigational technique in order to carry out the systematic review of developed approaches including the hybrid approach also. The several computational intelligence techniques have discussed below.

2.3.2.1 Genetic Algorithm

It is an evolutionary algorithm developed initially by Bremermann [74] in 1958, but the actual identity is given by Holland [75] in 1975 for firstly using the application of it in the field of computer science. It is the evolutionary computational approach used for the optimization of a selected function for maximization and minimization operation. It follows the biotic process of reproduction and natural selection mechanism to find out the fittest solution. In the process of evolution, the randomness present in the GA is to set by one to control the randomization. The genetic algorithm is more efficient and robust than random search algorithm without requiring extra information about the given problem. This key feature of GA is helping to get solution easily where all other optimization tools cannot handle due to deficiency of continuity and linearity. Genetic algorithm (GA) is a meta-heuristic algorithm and is used as powerful tool for optimization problem of all kinds. The ability to get global optima and high parallelism make it efficient over the other optimization technique. In 1975, Holland [75] has given the first introduction to GA for optimization problem based on Darwin's theory of survival of fittest. After that, it has been widely adopted in the field of mobile robotics. Shibata et al. [76] have proposed the successful GA based motion planning approach for the static environment in the presence of a polygonal obstacle. In the same year, Shing et al. [77] demonstrate the mobile robot for real-time path planning in planner terrain by adopting GA-based search strategy. Optimal path length, path smoothness and obstacle avoidance are the goals of effective path planning strategy proven by the Xiao et al. [78] by using GA in an unknown environment. Kang et al. [79] presented the solution to dead end problem of navigation of robot. They gave the GA to keep safe the robot from the stuck problem in the complex-crowded environment. When the robot falls in stuck like condition, then it operates online training to find a fit chromosome that helps the robot to come out from the situation. The path planning strategy in the presence of moving an obstacle in the unknown environment is presented by the Shi et al. [80]. The acceptability of GA is increased day by day and it also used with many other approaches to get hybrid approach for path planning. Pratihar et al. [81] have developed the fuzzy-genetic hybrid navigational approach for mobile robot path planning. Fuzzy logic is used to avoid obstacle present in the path and GA is used for to optimize path during the navigation. Hui et al. [82] have presented the study of GA for mobile robot navigation in combination with the fuzzy logic, neural network and compared with the conventional approach like potential field method. They observed that the fuzzy genetic and fuzzy neural approaches were working efficiently in terms of time optimality over potential field method. They also noted that the genetic algorithm with the fuzzy and neural network is more robust and adaptive as compared to potential field method. Wang et al. [83] applied the GA with PSO as a hybrid useful tool for robot path planning operation in the welding process. The hybrid mechanism enhances the diversity of particle and global search ability which makes collision free welding path and improves the welding efficiency as a result. The successive application of GA for the combinatorial optimization problem is provided by Kala [84]. He developed the multi-robot path planning over local search in limited computational infrastructure. Similarly, multiple goals optimization problems by GA are solved by Liu et al. [85]. In this approach, the energy consumption and idle time are minimized by implementing the new operator like repair, deletion and cut. Kuyucu et al. [86] improved GA for MRN over large search space to achieve the multi-objective task by using the combination of various mechanisms like genetic transposition inspired seeding, a strongly-typed crossover and multi-objective optimization. The path planning strategy in the presence of moving obstacle in the unknown environment is presented by the Shi et al. [87]. On the other hand the Yang et al. [88] showed the navigation of multiple mobile robots in the dynamic environment. Many researchers made some modification in conventional GA due to slow convergence rate, lack of cooperation between the population and local optimum. Hong et al. [89] presented the improved version of GA for global path planning of mobile robot. They used the co-evaluation mechanism for cooperation among the population of multiple mobile robots which results in avoidance of collision between robots and to gets optimal path during navigation. They demonstrated the present GA algorithm on multiple mobile robots in the presence of various static obstacles and showed better result over the basic GA. Carlos et al. [90] demonstrated the new form of GA for traveling salesman problem by considering dynamic target with respect to time. They achieved the goal by using simple prediction method and found the near optimal solution. Karami et al. [91] developed the adaptive genetic algorithm to present effective motion planning in 2D environment. In this work, they replaced the conventional selection operator by the adaptive one which continuously checks the fitness of the individual one and prevents the process from the premature convergence. It results in maintaining the diversity of the population in the solution and gives the quality of the solution.

2.3.2.2 Fuzzy Logic

The concept of fuzzy logic has been firstly given by Zadeh [92] in 1965 which is used in almost all field of research and development where to include a high degree of uncertainty, complexity and nonlinearity. The pattern recognition, automatic control, decision making, data classification are few of them. The theory of fuzzy logic systems is inspired by the remarkable human capacity to reason with perception-based information.

Rule-based fuzzy logic provides a formal methodology for linguistic rules resulting from reasoning and decision making with uncertain and imprecise information. In fuzzy behavior-based navigation, the problem is decomposed into simpler tasks (independent behaviors) and each behavior is composed of a set of fuzzy logic rule statements aimed at achieving a well-defined set of objectives.

- Fuzzy Logic used for modelling uncertain systems by enabling common sense reasoning in decision-making in the lack of complete and accurate information.
- It enables the arrival of a definite conclusion based on input information, which is unclear, uncertain, noisy and imprecise.

In 1965, Zadeh [92] provided the solution for real life problem and the knowledge based decision-making process by developing the mathematical theory which is widely known by "Fuzzy logic." The mobile robot using fuzzy logic can accurately move in the uncertain environment using array of sensory data and able to take its decision [93]. The collected sensory information plays a vital role to avoid the obstacle and building environmental map. The exact environmental map is used for the point to point navigation, robot localization, landmark identification and path planning. For getting effective navigation, a map building technique is used to learn the environment from the facts incrementally.

Robot behavior is implemented by using a set of fuzzy rules which combines the numerical data from sensors and linguistic data from the human experts [94]. If-Then rules [95-96] and inference engine is the main component of the fuzzy logic controller which encodes the mobile robots behavior. Zavlangas et al. [97] have presented the fuzzy based controller for obstacle avoidance. For the navigational purpose, Sugeno fuzzy based system with triangular and trapezoidal function is utilized for the omnidirectional mobile robot. Efficient fuzzy rules generation is the biggest problem of navigation because it requires the expert knowledge and human interference. Castellano et al. [98] have presented the automatic fuzzy rule generation by using human learning technique and machine learning technique. The algorithm such as genetic [99] and neural network [100] is used for automatic generation of the rules. Recently, fuzzy based navigation methodology is successfully dealing with the problem of static [101] and dynamic unstructured environment [102] by avoiding continuous making of loops and backtracking. Motlagh et al. [103] have developed fuzzy control technique for reactive navigation of a mobile robot to overcome the situation like dead end traps (U-shaped, maze, snails) and the getting trap in loop. Park et al. [104] presented the two fuzzy controllers strategy over complex environment. The first controller follows the general way of obstacle avoidance, target seeking and wall following while the other controller guides the robot to escape from the trap like a situation (U shape). Problems like navigation from the narrow passages have truly solved by Huq et al. [105]. Moustris et al. [106] proposed the path planner for curved trajectory by using the fuzzy logic. The designed fuzzy controller works efficiently in minimum rules. Fuzzy logic has been used in the combination with the sensor based navigation technique [107-110] to improve the incremental learning of the new environment. Parasuraman [111] presented the sensor fusion technique to improve the navigational rules to achieve realistic job using a modified fuzzy associative memory. According to him, while robot navigation in the complex-crowded environment requires large input space to match the environmental data and it also needed to optimize the number of rules. The proposed modified fuzzy associative memory provides the multiple input space and reduction in rules. Flanagan et al. [112] presented the Subsumption approach based on fuzzy rule mechanism for the wheeled mobile robot and the early model of the environment is not required for navigation. It follows the principle of "sense-plan-act" which allow the robot to generate behaviors. The behavior is a sudden response of the robot, such as "if obstacle present in a path then the motors get back the wheel". According to them, for obstacle avoidance behavior, it needed to detect the shadow appearing in the sensory area and details of what ahead are not so important. Recently, Fuzzy triangulation method [113] and reinforced based navigation [114] have been developed which helps to minimize the angular uncertainty and radial uncertainty present in the environment. Like sensor based navigation technique, the fuzzy logic has been successfully implemented with algorithm based navigation technique like neural network [115], genetic algorithm [116], potential field method [117] and many more in order to achieve an optimal perception of the environment which enables the robot to manage dead end situation. The problem of mobile robot navigation for the dynamic environment has been solved by Khatib et al. [118] and Lee et al. [119] by introducing fuzzy as data-driven approach for solving motion planning problem in the presence of the moving obstacle. Hoy et al. [120] have presented the cooperative approach for navigation of multiple mobile robots in unknown cluttered environment. They achieved the successful navigation task by using the fuzzy based auxiliary controller in limited sensing and communication capabilities for static environment. To improve the capability of the robot in moving condition Kang et al. [121] and Al-Mutib et al. [122] presented stereovision-based mechanism with fuzzy logic. To track the moving object, Abadi et al. [123] have designed the Mamdani based fuzzy controller for the wheeled mobile robot. They used particle swarm optimization algorithm with fuzzy logic as a hybrid approach to select the best parameters. The effective functioning of fuzzy logic has been presented by Castillo et al. [124] to maintain the diversity control in ant colony algorithm and to avoid premature convergence. They used the fuzzy approach to monitoring the unicycle mobile robot trajectory. Toloue et al. [125] have proposed the application of type-2 fuzzy with neural systems as a hybrid approach for parallel robots to handle the uncertainty of higher level. They provided highly accurate and low computational cost solution for position control of 3-Prismatic-spherical-prismatic parallel robot as compared to conventional approaches. The developed method omits the node pruning process and preserves the valuable rules when they needed. Rami et al. [126] presented the path planning strategy for multiple mobile robot systems and active motion coordination between them by using a probabilistic fuzzy controller with the neural network. In this approach, the leader robot position will follow by the follower robot. The first order Sugeno fuzzy system applied to head robot in order get high-level controller whereas companion robot has the low-level controller. The learning strategy is developed by using the neural network and efficient fuzzy rules are tuned by ANFIS. Fu et al. [127] have used the double layer fuzzy logic controller to minimize the complexity of the environment by actually controlling the heading angle and robots speed. The “Virtual target” and fuzzy based path planning strategy combined for path planning of mobile robot in the static and dynamic environment and result are shown in the simulational environment.

2.3.2.3 Firefly Algorithm

Yang [128] introduced Firefly algorithm (FA) in 2008. It is inspired by the fireflies flashing behavior, although it referred as meta-heuristics algorithm also. Its principle comprises to randomness states and general identification as trial and error of fireflies which is existing in nature stochastically. Firefly is winged beetle of family Lampyridae and commonly called as lightning bugs due to their ability to produce light. It produces light by a process of oxidation of Luciferin in the presence of the enzymes Luciferase, which occurs very quickly. This process of producing light is known as bioluminescence and firefly’s uses this light to glow without wasting of heat energy. Fireflies use this light for the purpose of selection of the mate, communicate a particular message and sometimes it also use for the scaring off animals who try to eat firefly. Recently the FA has been used as optimization tool and its application is spreading in almost all areas of science and engineering. The ability to self-plan, self-adaptation, and self-organize of FA is used by many researcher for various optimization problem such as fault detection in robot [129], economic emission dispatched problem [130], reliability-redundancy optimization [131], mixed variable structural optimization problem [132], cooperative networking problem [133], combinatorial optimization problem [134], learning from demonstration problem [135-136], dynamic environment problem [137-138] and many more. To improve the effectiveness of the firefly algorithm, some researcher used the Gaussian distribution function to increase the convergence speed [139] and some researcher modified the firefly algorithm to avoid random moving of the firefly algorithm when there is no brighter firefly [140]. Firefly algorithm is very efficient due to their ability to search for an optimal solution which is required for the solving science and engineering problem [141-143]. Yang [144] presented the novel approach based on firefly algorithm for solving the multi-objective problem by considering the nonlinear constraints. Baykasoglu et al. [145] presented the solution for real life problem of the dynamic environment by using firefly algorithm. The proposed work performs well when compared with genetic algorithm and differential evolution to solve the multidimensional knapsack problems for the static and dynamic environment. Due to the effectiveness of the firefly algorithm, it has been used with in combination with other algorithms as a hybrid approach. The Alweshah et al. [146] have presented the hybridized approach for solving the classification problem. They have proposed the firefly algorithm with simulated annealing, firefly algorithm with Levy flight and firefly algorithm with a simulated annealing-levy flight to create the improved balance between exploration and exploitation in obtaining a near-optimal weight for the probabilistic neural network. It results in high convergence speed for classification problem and maximizes classification accuracy. Maheshwar et al. [147] have improved the performance of the genetic algorithm by using firefly algorithm. The firefly algorithm contributed in the process of generation of the population of the chromosome and this hybrid approach results in global optimization at initial state and saved the system from the local minima. Zouache et al. [148] presented the new version of firefly algorithm with Particle swarm optimization algorithm as a hybrid approach to solve the continuous optimization problem. The non-linear approach problem is solved by Beykasoglu et al. [149]. They used the adaptive firefly algorithm for mechanical design optimization problems. To provide the solution for a non-linear problem, chaotic map with firefly algorithm is used. Nowadays, the work in the field of robotics by using FA started for mobile robot navigation. Hidalgo-Paniagua et al. [150] have presented firefly algorithm based mobile robot navigational approach in the presence of the static obstacle. They have achieved the three primary objectives of navigation such as path length, path smoothness and path safety. Brand et al. [151] presented the firefly algorithm for shortest and collision-free path for single mobile robot navigation in a simulational environment only. Sutantyon et al. [152] showed the application of firefly algorithm for the underwater mobile robot. They developed the scheduling strategy for swarm robots to avoid interference and jamming in marine conditioning. Firefly algorithm is used for programming strategy consist of phase synchronization algorithm. Sutantyo et al. [153] have solved the real world underwater navigation problem in the partial knowledge of the environment by using leavy flight- firefly based approach. The first firefly algorithm based cooperative strategy for detection of dead robots in multi mobile robot system by the active robot by synchrony is presented by Christensen et al. [154]. To solve the problem of aerial navigation Wang et al. [155] presented the new form of firefly algorithm for path planning of uninhabited combat air vehicle (UCAV). They developed the modified firefly algorithm to avoid the threat areas and to minimize the fuel cost during navigation of UCAV in the complicated battle field environment.

2.3.2.4 Neural Network

Artificial neural network is an intelligent system which consists of many simple and highly interconnected processing elements. These elements transfer the information by their capability of dynamic state response to external inputs. Neural network are basically shown by well-organized layers of interconnected nodes. The nodes consists the activation function. The input layer of neural network mechanism shown below in Figure 2.9 recognizes the patterns. These patterns then communicate to hidden layers for actual processing via a system of weighted connections. The hidden layers connect with the output layer to give the required answer.

The characteristics such as generalization ability, massive parallelism, distributed representation, learning ability, fault tolerance of neural network responsible for using in the field of mobile robot navigation. Janglova [156] presented the application of neural network for wheeled mobile robot navigation in the partially unknown environment. He used two neural network based mechanism for the development of collision-free path. The first neural mechanism finds the free space using sensory data and another neural network finds the safe trajectory by avoiding the nearer obstacle. Siemiatkowska [157] has presented the cellular neural network-based navigation strategy in the partially structured environment for mobile robot navigation. The first layer of cellular network deals with the signal sent by the map cells and neurons corresponds to the goal position and the current position is activated to reach real neurons. The second layer of the cellular network finds downhill search for the shortest path from robot to goal.

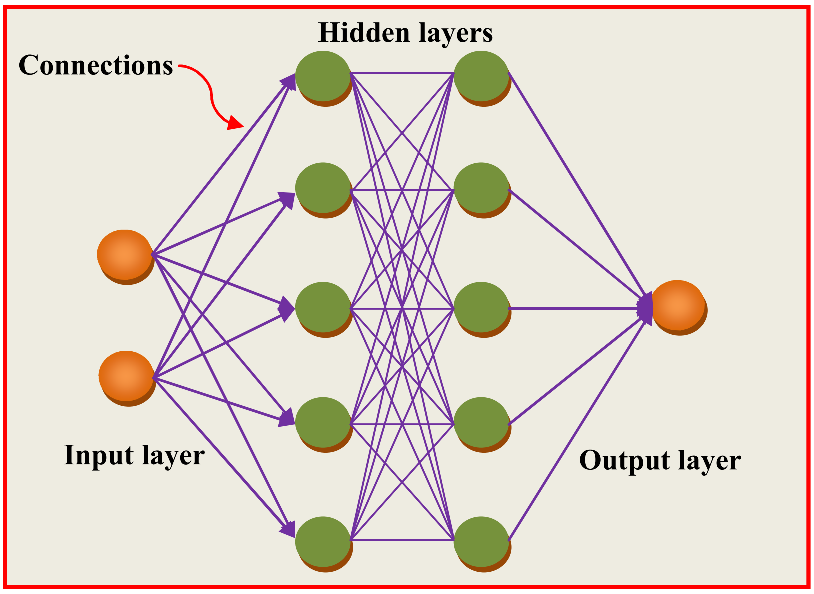


Figure 2.9: Architecture of neural network

Wang et al. [158] have presented the third generation artificial neural network i.e. spiking neural network (SNN) for obstacle avoidance. SNN generates the most believable models of the real biological neuron as compared to classical one. It has been applied easily to solve the non-linear classifying problem and high dimensional cluster. In order to avoid the human guidance in the process of the navigation, the Qiao et al. [159] presented the automation learning strategy. The feature of their work is that according to the complexity of the environment the neural network adjusts insertion and deletion of new hidden layers during the training without human guidance to accomplish the navigation task. Li et al. [160] have presented the application of neural network for Fast Simultaneous Localization and Mapping technique (Fast SLAM) in order to eliminate the error accumulation produced by the incorrect odometry model and inaccurate linearization of the SLAM nonlinear function. The use of the neural network with Fast SLAM enhances the mobile robot to navigate without collision with the obstacle in the unknown environment. To achieve the optimality during robot navigation operation, various approaches is used along with the neural network as a hybrid mechanism. Yong-Kyun et al. [161] have presented the combined effort of the neural network with potential field method to get cooperative coordination and competitive coordination for behavior-based control. Neural network classifies the environment based on sixteen prototypes of the topological map to describe the local navigational environment whereas the Potential field method selects the desired behavior. The proposed work gives a solution to the robot to escape out from the trapping situation by U-turn mechanism as the result of the hybrid approach. Pothal et al. [162] have presented the hybrid approach of the neural network and fuzzy to take the benefits of the both intelligent mechanism for multiple mobile robot navigations in clutter terrain. The proposed work is analyzed in the presence of the static obstacle. Like other Artificial intelligence technique, the neural network has been implemented with the sensor based method for mobile robot navigation. AbuBaker [163] has presented the novel hybrid approach for mobile robot navigation by combining the fuzzy logic with the neural network. The neural network effectively finds the optimum no. of activation rules to reduce computational for real time application. Pal et al. [164] have presented the application of neural network with sonar for mobile robot navigation. Medina–Santiago [165] have introduced the neural control system for the mobile robot in real time by using the ultrasonic sensors and Capi et al. [166] have presented the real-time navigation in urban areas by using the neural network with the help of GPS and compass sensor. To improve the performance of the neural network, the Syed et al. [167] have modified the basic neural network to form GAPCNN to get fast convergence of parameter for mobile robot while moving in the static and dynamic environment. The presented approach is modified by applying directional autowave control and accelerated firing of neuron based on the dynamic thresholding strategy. While working in the environment, it is essential for the robot to understand the complete dynamics. Hence, the Hendzel [168] presented the neural net motion planner which avoids the convex obstacle by using the neural network and also combines the behavior of ‘reaching the middle of the collision-free space’, ‘wall following’ and ‘goal seeking’. The filtered error approach is used to derive neural network controller. The self-learning strategy for the mobile robot is presented by Markoski et al. [169] is based on neural network. The pattern recognition is used as a tool for mobile robot navigation in the unknown environment with neural network and is demonstrated by Quinonez et al. [170].

2.3.2.5 Particle Swarm Optimization

Particle swarm algorithm (PSO) is nature based metaheuristic algorithm which adopts the social behaviour of the the animal likes fish scooling and bird flocking. It is developed by Eberhart and Kennedy [171] in 1995 and it is rapidly growing optimization tool for solving the various problems of the engineering and science. As the PSO mimics the behaviour of the social animal but they don't require any leader within group to reach the target. When the flock of birds goes to find the food they don't requires the leaders ulthough they go with one of the member who is at nearest position with food. In this way, the flock of birds gets their required solution by proper communication with the members of population. PSO algorithm consists of a group of particles where each particle represents a potential solution. Nowadays, PSO is widely used in the field of mobile robot navigation. Tang X. et al. [172] addressed the mapping and localization issues of mobile robot navigation in the unknown environment by using multi-agent particle filter. The use of PSO helps to minimize the calculation and holds more stable convergence characteristics. To get accurate trajectory and to save from trapping in local optima Xuan et al [173] have used PSO algorithm with MADS algorithm (Mesh Adaptive Direct Search). By using PSO_MADS algorithm together gives efficient result over the GA and EKF (Extended Kalman Filter). Atyabi et al. [174] have developed the Area Extended PSO (AEPSO) as the extension of basic PSO to address the dynamic and time-dependent constraint problems of mobile robot navigation. The AEPSO approach is successfully implemented in search of survivor rescuing and bomb disarming. To handle the navigation of multi mobile robot system, the Tang et al. [175] have addressed cooperative motion path planning in the complicated environment by using the PSO. The PSO in combination with the multibody system dynamics consisting of the properties of robot like acceleration, mass, force, inertia which is then considered for investigation of fault tolerance of the proposed approach. The some modification have been made by Couceiro et al. [176] for navigation of multiple mobile robots in the real world. They modified form of PSO and Darwinian PSO (DPSO) system for obstacle avoidance and mutual communication issues. They found that in the system of 12 physical robots the achieved efficiency upto 90% in a sense of maximum communication distance and global optimum. Chen et al. [177] have tried to develop the human expert control strategy with learning based ability for the uncertain environment by using multi-category classifier. For this, PSO is used to get higher accuracy within the short time. On comparison with the conventional grid search, it has been noticed that it has higher classification accuracy without prematurity. To develop the efficient path planning, a hybrid approach has been given by Das et al. [178]. They presented the application of PSO and improved gravitational search algorithm (IGSA)as a hybrid methodology to evaluate the optimal path planning for multiple mobile robots in the clutter environment. The hybrid approach makes balance between exploitation and exploration by adopting the co-evolutionary technique which update IGSA accleration and PSO velocity. The application of PSO for the underwater robot is developed by He et al. [179]. They have proposed the PSO-UFastSLAM approach to get the improvement in accuracy of estimation and to limit the particle to get better results. Algabri M. et al. [180] presented the comparative study of the GA, PSO, the neural network with fuzzy logic and they observed that the fuzzy logic paired with PSO gives the right results concerning distance travelled.

2.3.2.6 Ant Colony Algorithm

Ant colony optimization (ACO) is a swarm intelligence algorithm developed by the Marco Dorigo in 1992 in his Ph.D. thesis [181]. It is the population-based approach used to solve the combinatorial optimization problem. ACO algorithm is originated from the behavior of ants and its ability to find the shortest path from their nest to food source. ACO algorithm is already applied to various fields of science and engineering such as job-shop scheduling problem, vehicle routing problem, assignment problem, set problem and much more. Nowadays, ACO is used to handle the mobile robot navigation problem for obstacle avoidance and effective path planning due to its ability to tackle the real-time problem. Guan-Zheng et al. [182] presented the application of ACO for real-time path planning of mobile robots. The adoption of ACO increases the convergence speed, solution variation, computational efficiency and dynamic convergence behavior when compared with other algorithms like GA. The navigation for multiple mobile robots is presented by Liu et al. [183] by using ACO. They presented collision avoidance strategy for various robot systems in the static environment. They used the special function to improve the selective strategy. When the ant finds dead- corner then penalty function is used for the trail intensity updated to avoid the path deadlock of the robot. The hybrid approach for mobile robot navigation is presented by Castillo et al. [184]. The combined effect of ACO and fuzzy logic is taken. The fuzzy logic approach plays key roles in diversity control in the ACO. The primary objective of the proposed approach is to avoid full convergence through dynamic variation of a particular parameter. Purian et al. [185] have presented the application of ACO algorithm for mobile robot navigation in the unknown dynamic environment. They have used the ACO for selection and optimization of the fuzzy rules.

2.3.2.7 Other Miscellaneous Algorithm

Many researchers developed the different nature inspired techniques to perform the successful navigational task for mobile robot such as Cuckoo Search (CS) Algorithm [186], Bacterial Forging Optimization (BFO) [187-189], Artificial Bee Colony (ABC) algorithm [190-191], Invasive Weed Optimization (IWO) [192], Shuffled Frog Leaping Algorithm (SFLA) [193-194], Bat Algorithm (BA) [195].

2.4 Discussion

The rigorous literature review on mobile robot navigation is carried out and classified according to their nature. It is observed that the implementation of AI based navigational algorithm is preferred as compare to classical approaches. The applicability and high computational capacity prefer selection of AI based navigational approaches for path planning of mobile robot. Nowadays, nearly 90% of the research is being done in the development of the AI-based approaches however the 10% work is going on by using classical approaches. The Figure 2.10 shows the year wise work carried out in the field of mobile robot navigation by using classical approaches and AI approaches. From the bar graph, it is clear that the application of the classical approaches for mobile robot navigation is decreasing decade by decade and on the other hand the application of the AI-based approaches is increasing decade by decade. The doughnut shown in Figure 2.11 clears the contribution of AI approaches such as fuzzy logic, genetic algorithm and neural network is more as compare to other AI approaches like PSO, ACO, FA and Miscellaneous algorithm. The application of firefly algorithm and hybrid approaches is very limited for mobile robot navigation.

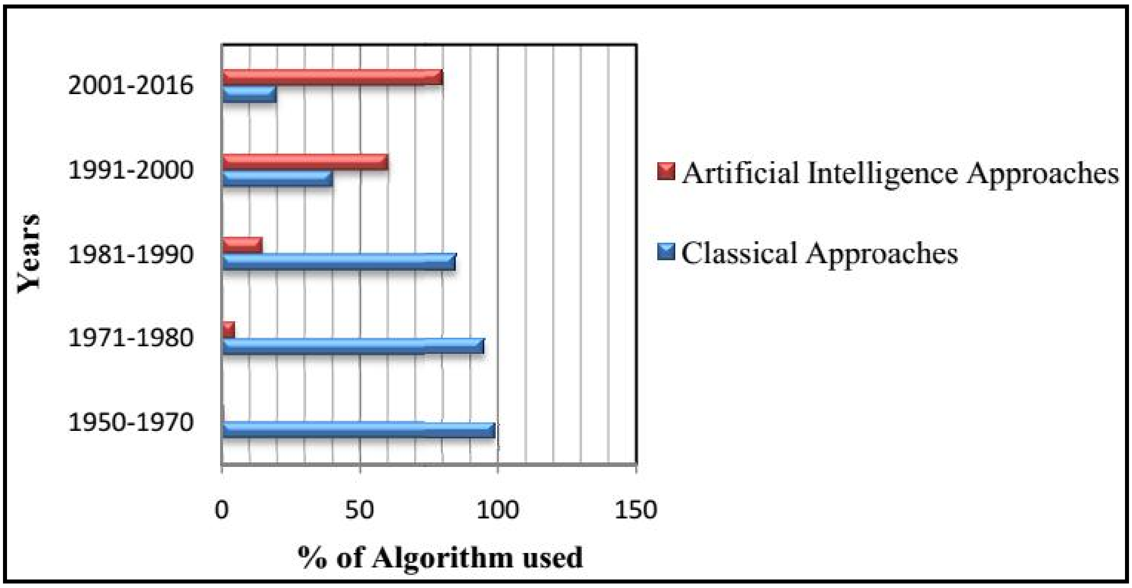


Figure 2.10: Development of mobile robot navigation approaches

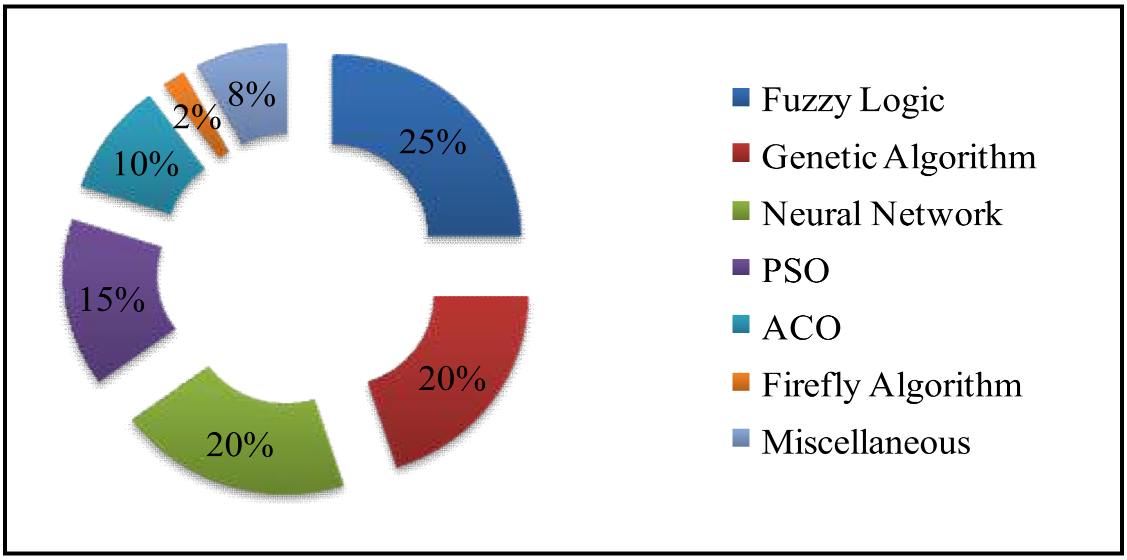


Figure 2.11: Percentage of paper reviewed on mobile robot navigation using AI approaches

2.5 Summary

After the complete analysis of the literature, the following key points are emerged and are given below:

- Mobile robot navigation in the unknown or partially unknown environment is successfully done by various classical and AI-based approaches.
- The AI based navigational approach for mobile robot navigation is preferred as compared to classical approaches.

- The research on mobile robot navigation by using the nature-inspired algorithm such as FA is very limited for path planning in complex and unknown environments.

The proposed research work gives the new navigational approach such as PFL (Probability-Fuzzy Logic) Controller, MGA (Matrix based Genetic Algorithm), FA (Firefly algorithm), FA-PFL (Firefly Algorithm-Probability-Fuzzy Logic), FA-MGA (Firefly Algorithm-Matrix based Genetic Algorithm), FA-PFL-MGA (Firefly Algorithm-Probability-Fuzzy Logic-Matrix based genetic algorithm) hybrid approaches for carrying out navigational path planning for single and multiple mobile robots.

Chapter 3

Kinematics of Wheeled Mobile Robot

The chapter provides the kinematic analysis of the wheeled locomotion mechanism for the wheeled mobile robot (WMR). Wheeled locomotion is popularly being used for mobile robots especially for a situation like high risk and rough terrain. Such situation may result in loss of vehicle stability, wheel traction and controllability. Therefore, it is essential to study the kinematics of the automobile robot. It deals with the study of the mechanical behavior of the robot without considering the effect of force and to develop control software to attain thorough command over the hardware of the mobile robot.

3.1 Introduction

Kinematics plays an important role in the behavior of the WMR during navigation. It is the fundamental study of the mechanical behavior of the WMR without considering the force. The mechanical behavior of the system and the control software to handle the hardware is the important issues while designing the robot for a particular application. In the autonomous navigation of WMR, kinematics study is essential because the robot has to move continuously in its environment. The control system plays an important role in planning the trajectory which is followed by a robot in a specific direction. Therefore, for successful navigation, the localization of its current position in its frame of reference is an important issue of discussion. For effective navigation of WMR, the selection of the wheel on which it kept and its geometric constraint plays a vital role. For deriving model, it is required to decouple each part of the robot so that the parts can be analyzed separately. Development of the model is a bottom-up method in which every single wheel contributes to the robot movement and enforces the constraints on robot’s motion at the same time. The connection between the robot chassis and plane surface is made by using some wheels and hence the wheel is the rigid link between them. Therefore, the robot motion depends on the wheel and its constraint. A reference frame is thus essential to express the forces and constraints of each wheel. Due to mobile nature of the robot, it is important to have an ability to map the environment between the global and the local reference frames.

3.2 Model of the System

The following assumption has been considered throughout the analysis:

- The WMR is assumed as rigid body and navigation is considered on the plane surface.
- The dimensionality of WMR is three, two along the horizontal plane and one along the vertical plane.
- The effects of joints on the robot chassis and the internal degree of freedom are neglected.

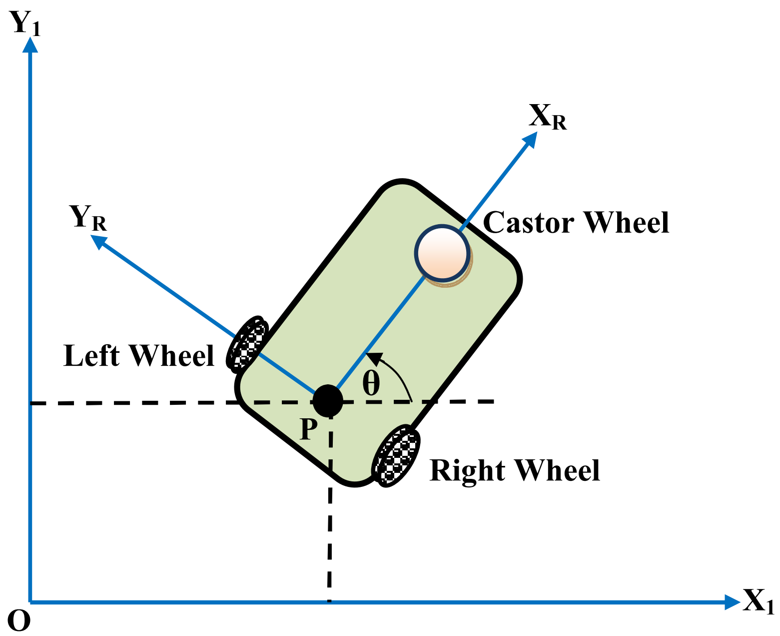


Figure 3.1: Model of the WMR

In Figure 3.1, the WMR is shown along with its reference plane PX_RY_R and the environment where it localizes its position with respect to plane OX_1Y_1. The positioning of the robot in its environmental frame OX_1Y_1 is presented by the equation 3.1.

$$\xi = [x \quad y \quad \theta] \tag{3.1}$$

Where x and y represents the coordinates of the point 'P' of the robot and θ represents the robot orientation and mapping between the two reference planes namely global and local is accomplished using the orthogonal rotation matrix:

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

The equation 3.3 presents the movement of the robot from one position to another in the environment. The matrix R(θ) shown by equation 3.2 can be used to map motion in the global reference frame $\{X_1, Y_1\}$ to motion in terms of the local reference frame $\{X_R, Y_R\}$. This operation is denoted by R(θ) $\dot{\xi}_I$ because the computation of this operation depends on the value of angle θ.

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I \tag{3.3}$$

3.3 Mobile Robot Wheel Constraints

The effect of the wheel constrains on the performance of the mobile robot is studied in this section. The kinematic model of various types of the wheel is presented. The assumption is considered for the kinematic analysis as follows:

- There must be point contact between surface and wheels.
- The wheel must be rigid.
- The rolling must be friction free.
- The wheel must avoid the skidding.
- The plane surface and steering action must be orthogonal.
- The connection between the wheels and chassis of the robot must be rigid.
- At the time of the motion, the wheels plane must be vertical and wheel must rotate along the horizontal axle.

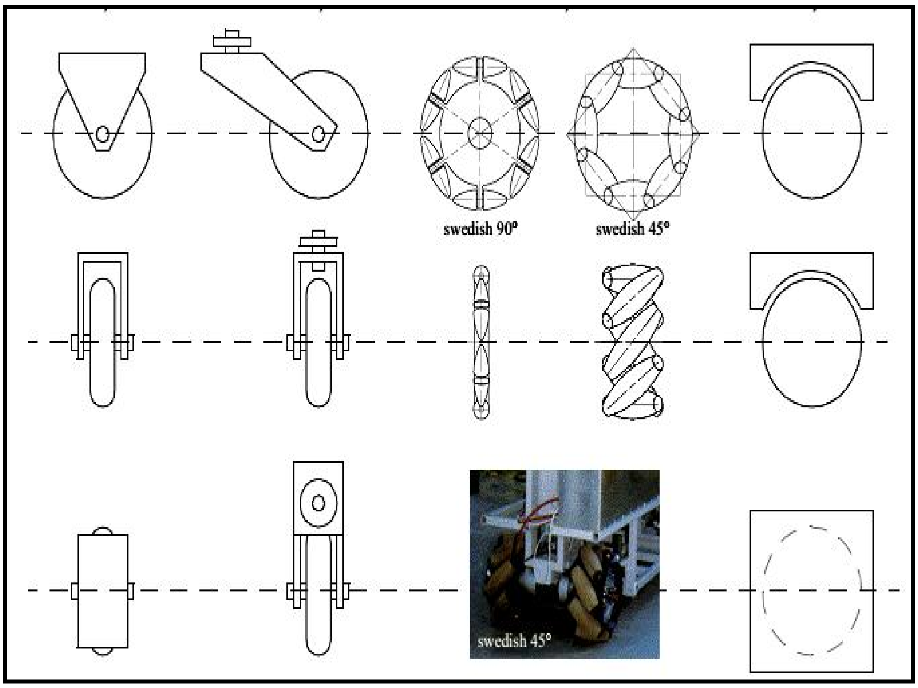


Figure 3.2: Various types of wheel mechanism for MRN

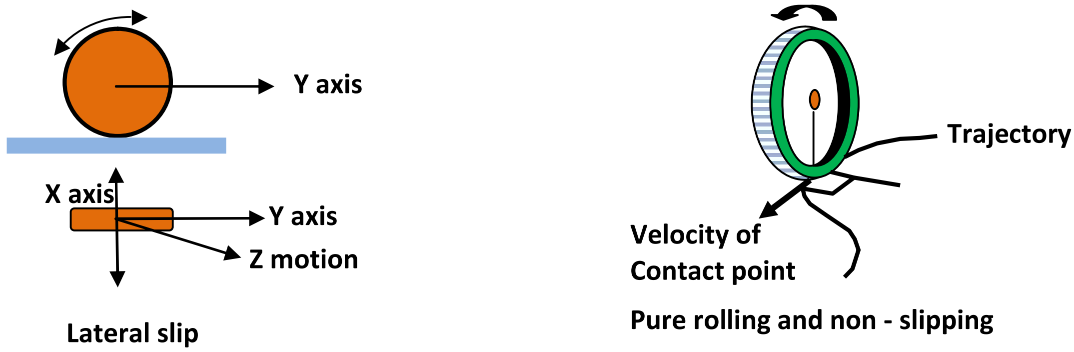


Figure 3.3: WMR Kinematic constraints (a) Pure rolling (b) Lateral sleeping

3.4 Geometry of Wheels

The two kinematic constraints are considered for wheel analysis is shown in Figure 3.3. By geometry of wheels, it is classified as:

- Conventional wheel (Fixed standard wheel)
- Castor wheel
- Steered standard wheel
- Spherical wheel
- Swedish wheel

3.4.1 Conventional Wheel (Fixed Standard Wheel)

In a conventional wheel, the robot can move along only forward and backward direction in the plane. The rotation of the vertical axis is not possible due to fixed constrained. The Figure 3.4 shows a conventional wheel (Fixed standard wheel) A and indicates its position pose relative to the robot's local reference frame $\{X_R, Y_R\}$. The position of A is expressed in polar coordinates by distance l and angle α . The angle of the wheel plane relative to the chassis is denoted by β, which is fixed since the fixed standard wheel is not steerable. The wheel, which has radius r, can spin over time, and so its rotational position around its horizontal axle is a function of time t: $\varphi(t)$.

The equation 3.4 shows the pure rolling condition as:

$$[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad (-1)\cos\beta]R(\theta)\dot{\xi}_I - r\dot{\phi} = 0 \tag{3.4}$$

Due to the sliding constraint, the wheel's motion becomes zero when observed normal to the wheel plane.

$$\begin{bmatrix}\cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta\end{bmatrix}R(\theta)\dot{\xi}_I = 0 \tag{3.5}$$

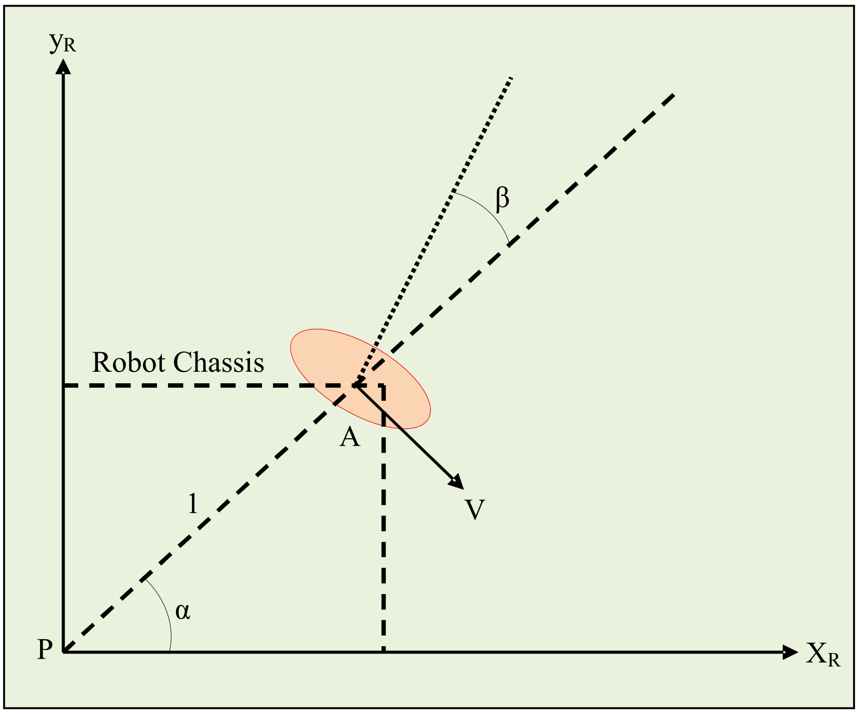


Figure 3.4: Geometry of the Conventional wheel

3.4.2 Steered Standard Wheel

Standard steered wheel is improved form of the conventional wheel. It gives a surplus degree of freedom compared to conventional wheels. The movement of the wheel around the vertical axis is possible in this case. The angle between the wheel and robot may change according to time in case of steered standard wheel.

The rolling and sliding constraints in standard steered wheel along the wheel plane are given as:

$$\begin{bmatrix}\sin(\alpha+\beta) & -\cos(\alpha+\beta) & (-l)\cos\beta\end{bmatrix}R(\theta)\dot{\xi}_I - r\dot{\phi} = 0 \tag{3.6}$$

And normal the wheel plane is

$$\begin{bmatrix}\cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta\end{bmatrix}R(\theta)\dot{\xi}_I = 0 \tag{3.7}$$

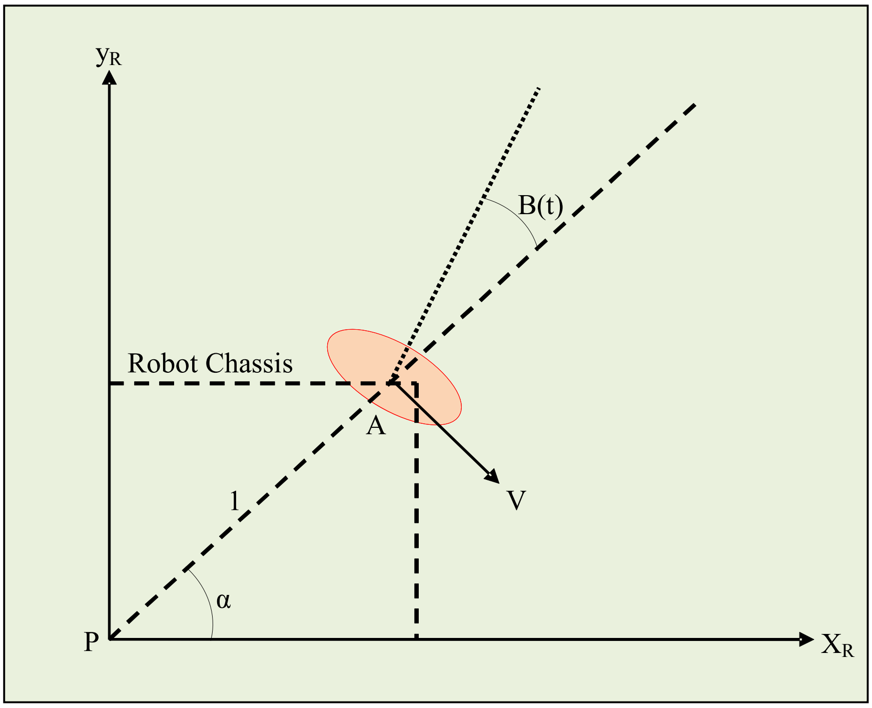


Figure 3.5: Geometry of Steered standard wheel

3.4.3 Caster Wheel

The castor wheel works like the steered standard wheel. It also has the forward and backward movement along a horizontal plane and rotating movement with the vertical axis, but the vertical axis of rotation does not pass through the contact point of the wheel and plain surface.Therefore, the special parameter *d* is linked with the wheel for analysis of the caster wheel as shown in Figure 3.6. Thus to specify the position of the caster wheel, one additional parameter (fixed length of rigid rod 'd') linked to the wheel, as shown in Figure 3.6.

The rolling and sliding constraints of Castor wheel along the wheel plane are given as:

$$\left[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad (-l)\cos\beta\right] R(\theta)\,\dot{\xi}_I - r\phi = 0 \tag{3.8}$$

Orthogonal to the wheel plane;

$$\left[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta\right] R(\theta)\,\dot{\xi}_I + d\dot{\beta} = 0 \tag{3.9}$$

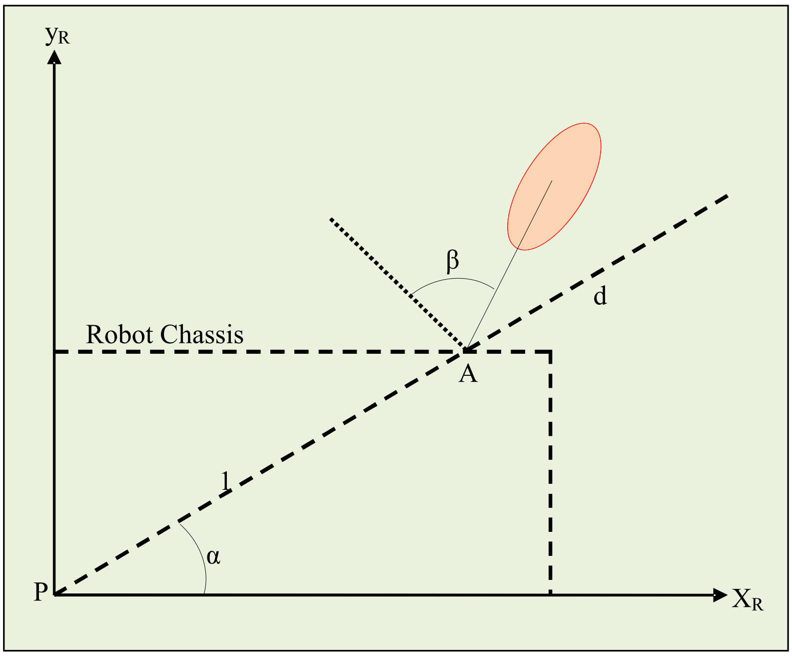


Figure 3.6: Geometry of Caster wheel

3.4.4 Swedish Wheel

The Swedish wheel is omnidirectional wheel as it has movement along all direction. It does not have the vertical axis of rotation like steered standard wheel and castor wheel. While designing the Swedish wheel minimum three rollers must require. The rollers are arranged on the periphery of the wheel in such a way that the axis of rollers is arranged tangent to wheel perimeter and free to rotate.

In Figure 3.7, the angle γ stands for the measured angle of the main axis with roller axes

The rolling and sliding constraints in Swedish wheel along the wheel plane are given as:

$$\left[\sin(\alpha+\beta+\gamma) \quad -\cos(\alpha+\beta+\gamma) \quad (-l)\cos(\beta+\lambda)\right] R(\theta)\,\dot{\xi}_I - r\dot{\phi}\cos\gamma = 0 \tag{3.10}$$

Orthogonal to the wheel plane;

$$\left[\sin(\alpha+\beta+\gamma) \quad \cos(\alpha+\beta+\gamma) \quad l\sin(\beta+\lambda)\right] R(\theta)\,\dot{\xi}_I - r\dot{\phi}\sin\gamma - r_{sw}\,\dot{\phi} = 0 \tag{3.11}$$

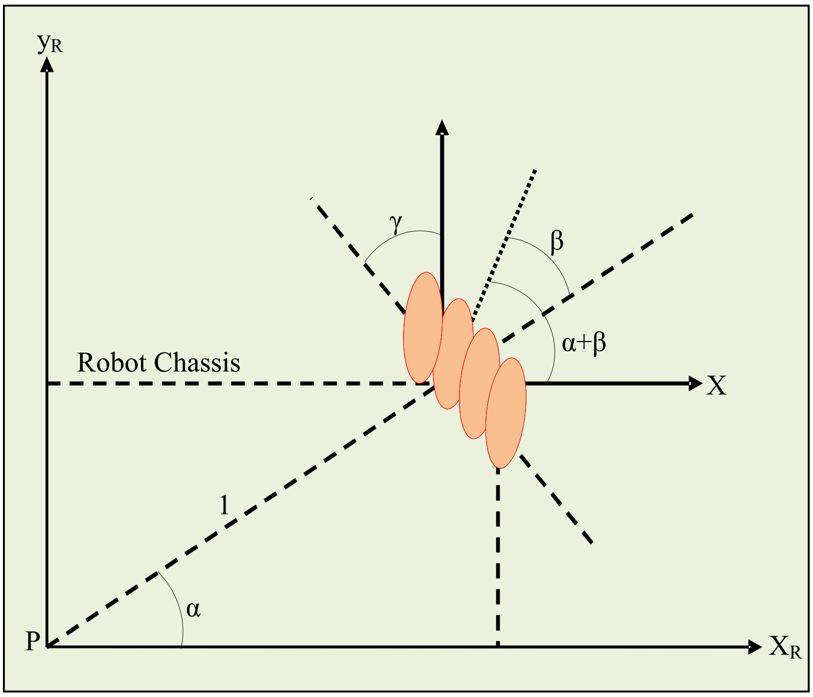


Figure 3.7: Geometry of Swedish wheel

3.4.5 Ball Wheel (Spherical Wheel)

The ball wheel or Spherical wheel is free to move in any direction as they have no constrained along any direction. The ball wheel is omnidirectional and they have no any kinematic constrained over robot chassis.

Figure 3.8 shows the analysis of the ball wheel at point A.

The rolling and sliding constraints in Ball wheel along the wheel plane are given as:

$$\left[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\phi} = 0 \tag{3.12}$$

Orthogonal to the wheel plane;

$$\left[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta\right] R(\theta)\dot{\xi}_I = 0 \tag{3.13}$$

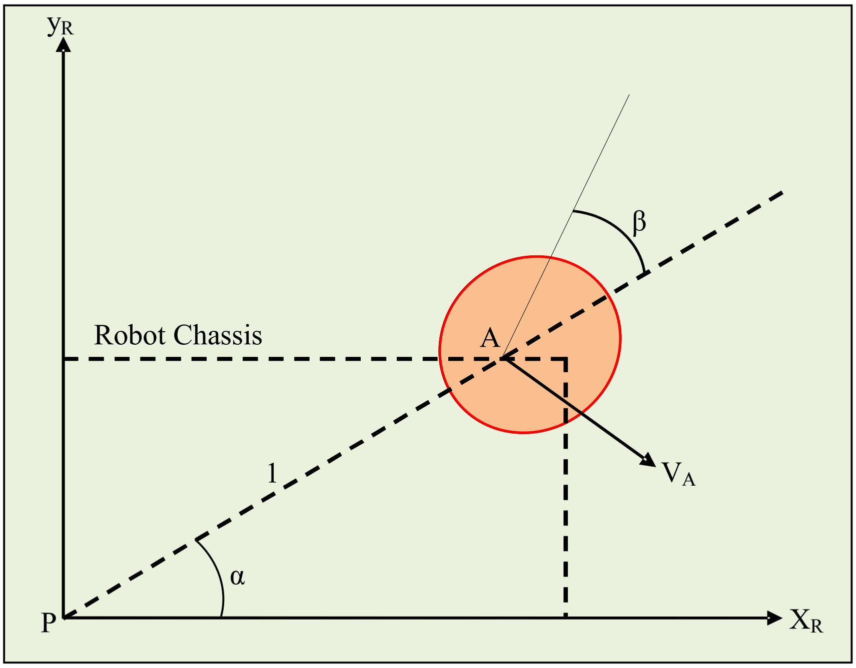


Figure 3.8: Geometry of Ball Wheel

3.5 Kinematic Constraints of the WMR

For analysis of the system, consider the system of WMR say *N,* which consists of Castor wheel (N_C), Spherical wheel (N_{SPH}), Swedish wheel (N_{SW}), fixed standard wheel (N_F) and steered standard wheels (N_S). The above section clears that the wheels such as N_C, N_{SPH} and N_{SW} have no kinematic constrained over the robot chassis while the wheels such as N_F and N_S have great influence on the chassis. Therefore, while considering the N_F and N_S for navigation, much attention requires on their kinematic constraints.

The equation 3.14 is the unique equation which allows the rolling constraints of all wheels.

$$J_1(\beta)R(\theta)\dot{\xi}_1 - J_2\dot{\phi} = 0 \tag{3.14}$$

Here, J_1 stands for matrix representation of the motion of all wheels.

J_2 *is* $N \times N$ diagonal matrix of all wheels radii.

$$J_1(\beta) = \left[J_1F(\beta)\ J_{1s}(\beta_s)\ J_{1c}(\beta_c)\ J_{1sw}\ J_{1sph}\right]^T \tag{3.15}$$

The matrix J_{1F}, J_{1S}, J_{1C}, J_{1SW} and J_{1SPH} are of size (N_F x 3), (N_S x 3), (N_C x 3), (N_{SW} x 3) and (N_{SPH} x 3) respectively. J_2 *is* $N \times N$ diagonal matrix of all wheels radii but in the case of the Spherical wheel, the term *cosγ* is multiplied.

The equation 3.16 combines the all constraints of the wheel shown as:

$$C_1(\beta)R(\theta)\dot{\xi}_1 = 0 \tag{3.16}$$

Where,

$$C_1(\beta)=\left[C_1F(\beta)\ C_{1s}(\beta_s)\ C_{1c}(\beta_c)\ C_{1sw}\ C_{1sph}\right]^T \tag{3.17}$$

3.6 Degree of Mobility of the WMR

The wheels such as Ball Wheel, Swedish wheel and Castor wheel has no effect on the robot chassis but the steered standard wheel and fixed standard wheel influences the constraints on the robot chassis. Therefore, the steered standard wheel and fixed standard wheel is considered for analysis.

Assume ($N_F + N_S$) wheels, to avoid lateral slip;

$$C_{1f}R(\theta)\dot{\xi}_1 = 0 \tag{3.18}$$

$$C_1(\beta_s)R(\theta)\dot{\xi}_1 = 0 \tag{3.19}$$

$$C_1(\beta_s) = \left[C_{1f} \quad C_{1s}(\beta_s)\right]^T \tag{3.20}$$

Mathematically, the null space of $C_1(\beta_s)$ is the space *N* such that for any vector *n* in *N*,

$$C_1(\beta_s).n = 0 \tag{3.21}$$

The steerable standard wheels and fixed standard wheels are avoided for practice. Therefore the probable range of rank values for any robot is $0 \le \left[C_{1s}(\beta_s)\right] \le 3$.

If the rank is $\left[C_{1s}(\beta_s)\right] = 0$, it means the zero independent kinematic constraints in $C_{1s}(\beta_s)$. In this case, fixed standard wheel and steerable is not attached to the robot. Therefore $N_F = N_S = 0$.

If the rank is $\left[C_{1s}(\beta_s)\right] = 3$, it means the constrained in all direction, therefore, motion is not possible.

In the following equation degree of mobility δ_m is formulated as:

$$\delta_m = \dim N\left[C_{1s}(\beta_s)\right] = 3 - \text{rank}\left[C_{1s}(\beta_s)\right] \tag{3.22}$$

3.7 Degree of Steerability:

It is the ability of the robot to get steer freely in the environment by using various wheels. It can be formulated as:

$$\delta_s = \mathrm{rank}\left[C_1(\beta_s)\right] \qquad (3.23)$$

The range of δ_s can be specified: $0 \le \delta_s \le 2$.

3.8 Robot Maneuverability

The Maneuverability of WMR represents the capability of the robot to provide smooth functioning during navigation. The smooth functioning of the robot is defined regarding the mobility by considering the robot kinematic constrained and an additional degree of freedom for steering. It is the combination of the movement of the robot chassis in its environment with respect to time and ability to localize its position in the global frame of reference.

The Maneuverability of the robot (δ_M) is presented in terms of its degree of mobility and steerability as:

$$\delta_M = \delta_m + \delta_s \qquad (3.24)$$

Robot maneuverability for three wheel configuration is given below in Table 3.1.

Table 3.1: Robot maneuverability (δ_M) for five basic types of three wheel robots

Sl. No.	Wheel Configuration	δ_m	δ_s	δ_M
1	Omnidirectional (Three Spherical wheels)	3	0	3
2	Differential (Two Fixed standard wheels and one Spherical wheel)	2	0	2
3	Omni-steer (Two spherical wheels and one Steered standard wheel)	2	1	3
4	Tricycle (Two Fixed standard wheel and one Steered standard wheel)	1	1	2
5	Two steer (two steered standard wheels and one spherical wheel)	1	2	3

In Table 3.1, we can replace the Swedish wheel and caster wheel by ball wheel, these changes will not affect the maneuverability of the wheel.

3.9 Kinematic Analysis of the Differential Drive WMR

For the analysis, the differential drive WMR is considered with its parameters as shown in Table 3.2.

Table 3.2: Parameters of the kinematic model of the mobile robot.

Sl. No.	Symbol	Parameter
1	r	Radius.
2	l	Distance between the two driving wheels along the y-axis robot.
3	V_R	Linear velocity of the right wheel.
4	V_L	Linear velocity of the left wheel.
5	V_ω	Angular velocity of the robot.
6	V_l	Linear velocity of the robot along x-axis of the robot.
7	c	Centre of the axis of the rear wheels.
8	R	Radius of curvature for the robot trajectory.

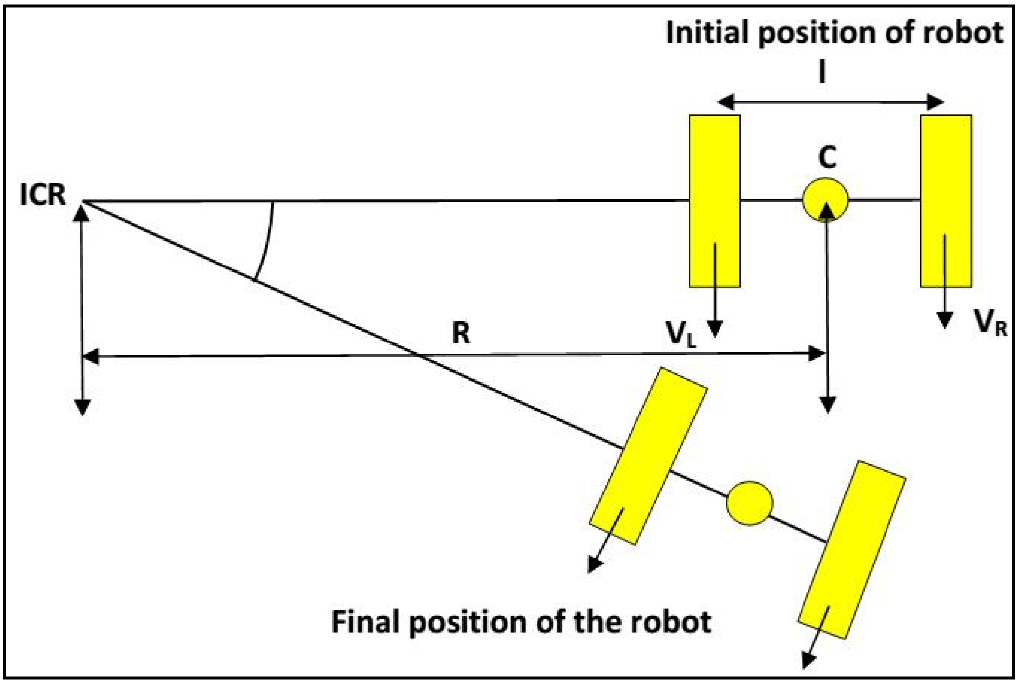


Figure 3.9: Instantaneous centre of rotation (ICR)

Let us consider, at any instance of time‘t’ the robot follows a path shown in Figure 3.9. The robot turns along a curve around the instantaneous center of rotation (ICR). Angular velocity will be given by;

$$V_\omega(t) = \frac{d\theta(t)}{dt} \tag{3.25}$$

Hence, the robots linear velocity is as:

$$V_l(t) = V_\omega(t)R(t) \tag{3.26}$$

Also

Left wheel velocity and right wheel velocity is expressed as;

$$V_L(t) = V_\omega(t)(R(t) + \frac{l}{2}) \tag{3.27 a}$$

$$V_R(t) = V_\omega(t)(R(t) - \frac{l}{2}) \tag{3.27 b}$$

On solving the above equations, radius of curvature of the robot trajectory will be given by;

$$R(t) = \frac{l(V_L(t) + V_R(t))}{2(V_L(t) - V_R(t))} \tag{3.28}$$

Now angular velocity and linear velocity of the robot can be rewritten as;

$$V_\omega(t) = \frac{V_L(t) - V_R(t)}{l} \tag{3.29 a}$$

$$V_l(t) = \frac{V_L(t) + V_R(t)}{2} \tag{3.29 b}$$

It is clear from the above equations that, by changing the velocities of the two driving wheels of the robot, different trajectories can be formed by the robot. In differential drive robot, small variations in the velocities of the two wheels result in significant change in the trajectory. Therefore, errors due to slippage should take into consideration during the trajectory planning of a mobile robot.

The kinematic equations in the world frame can be written as;

$$\dot{x}(t) = V_l(t)\cos\theta(t) \tag{3.30 a}$$

$$\dot{y}(t) = V_l(t)\sin\theta(t) \tag{3.30 b}$$

$$\dot{\theta}(t) = V_\omega(t) \tag{3.30 c}$$

On integrating the above equations, we get;

$$x(t) = \int_0^t V_l(\tau)\cos\theta(\tau)d\tau + x_0 \tag{3.31 a}$$

$$y(t) = \int_0^t V_l(\tau)\sin\theta(\tau)d\tau + y_0 \tag{3.31 b}$$

$$\theta(t) = \int_0^t V_\omega(\tau)d\tau + \theta_0 \tag{3.31 c}$$

Where (x_0, y_0, θ_0) represents the initial position of the robot. Above equations can be used for the robot capable of travelling towards a particular direction $\theta(t)$ with a velocity of $V_l(t)$.

3.10 Summary

The chapter presents the kinematic analysis of wheeled mobile robot which helps to understand the important factor which is responsible for the motion of the robot when working in its environment. The key finding of the work is as follows:

- The kinematic analysis of WMR helps to understand the kinematic constrained of the wheel which is necessary to understand for designing the robot for a specific task.
- It helps to localize the position of the robot in the global frame of reference during navigation.
- It gives the study of maneuverability for smooth motion of WMR based on the degree of mobility and steerability.
- The kinematic analysis gives the study over velocity of the individual wheel which directly controls the motion of the robot and generates the required trajectory.

The study of the kinematics of WMR helps for successful implementation of the AI techniques for robot navigation.

Chapter 4

Mobile Robot Navigation by Matrix based Genetic Algorithm

In the current research the finite bit string matrix is proposed for Mobile Robot Navigation. The Mobile Robot Navigation is based on matrix-binary codes representation of Genetic Algorithm. This representation transforms the spectrum of the objective functions of mobile robot navigation into the unique objective function up to the level of desired best fit. The random environment is defined by the matrix, which is represented by the coding as an array of the finite bit strings. The decision variables, objective function and the constraints are interacted in the Matrix Algebra. The optimum decision is transformed into the logic decision by the Matrix-Binary Codes used for navigation of mobile robot.

4.1 Introduction

The modern engineering is influenced with the optimization and its frontiers. Robotics control performance is bounded in the optimum interval as the heuristic environment and path decision. The heuristic environment is clustered into an array and this optimized path is found out using the best fit objective function. This chapter is structured by the genetic algorithm over matrix trace representation. The main challenges of navigation are obstacles avoidance and path planning of robot. The obstacles are represented by the binary codes and their position are represented by the matrix in order to get collision-free environment.

4.1.1 Genetic Algorithm Principles

Charles Darwin's Theory of Evolution becomes a potential tool to establish an intelligent searching algorithm. This modern application is known as Genetic Algorithm (GA). GA is an algorithm not only used for searching but extensively used for optimizing, simulating, selecting, eliminating, and optimization model to establish as a method of searching for filtering the decision up to optimization. In a biological colony, there is randomness in evaluation, selection and elimination. The random variables such as the evaluation period, the heredity selection and the natural elimination of biological colony represent by the objective function of MRN. There is the global probability in these variables but to search the best fit global probability is the desired objective. The biological analysis drafted into the sequential algorithm and its matrix representation reforms as a genetic algorithm. The chromosomes are considered as the elements of atoms and the genetics space as the molecules. These are consisting of the best fit objective function. The GA plays an important role in micro-macroscopic search and represents the global data. These data consists hidden and unknown information which is referred as the weights and GA fits these weights in the objective function under the best-fit conditions of MRN. Hence, GA performs as an optimized controller for the navigation. Genes are the decision variables in GA and its characteristics are refered as the biological restriction. This generated set constructs the objective function for fitting the best-fit gene. This objective function is studied over the subject to the constraint of distinct inequalities as population. Next, the evolution of adoptability is studied under the biological restrictions. It is a searching technique to set a best-fit population.

The GA consists of the following three biological characteristics:

- Selection
- Crossover
- Mutation

Each characteristic is formulated according to the objective function subject to the constraints of biological circumstances and non-negative restrictions. The Selection refers to the regeneration, reproduction and copying the genes. Each individual is studied under the fitness function. This calculation is proportional to select a better individual with better fitness. This is copied and then it is transferred for generating the new population. The elimination process runs simultaneously for those individuals, which has the low fitness function. Hence, the selection and elimination execute in parallel with the level of the fitness function. The new population and the new fitness function are the current surfaces before the succeeding steps. There are several existing methods of selection such as roulette wheel selection, cyclic selection, expected value selection, paired competition selection, retaining high-quality individual selection. The next process after the selection in GA is a crossover. Matching to the pool is the primary objective of crossover process. Match pool is defined as the relation of the selected best-fit individuals. Crossover reforms into pairs from the individuals. More precisely, to generate the parents, crossover plays an important role. Every individual of the environment existed randomly. By the crossover mechanism, it converts into well-defined pairs. Next, the probability is applied on the crossover for occurring the new pairs. To obtain the new genes for the next generation is the key objective of this probability application. Hence, these become the new individuals who also are represented as the stochastic variables. These are the set of information which analyzes under the condition of crossovers. This crossover may be classified into one point, multi-point and average-point crossover. In the process of mutation, the substitution of the genes with the opposite genes performs as per the rate of mutation. The position of the genes is traced and the new individual is to be fixed for the new cyclic process of this gene-based genetic system. Finally, the selection, crossover and mutation are being in that system. This system carries the best fit objective function over the stochastic genes for the optimized searching. The mathematical representation of GA is required to control the system. Its mathematical modeling is presented below:

4.2 Mathematical Modeling of GA

GA is based on Darwin's Theory of Evolution. The challenge of the existence of all living things was the key reason for this theory. The "survival of the fittest" is defined over the living things, where the new breeds come by the system of reproduction, crossover and mutation from the present organisms. GA is the transformation from natural to computational. First, by setting the problem then obtain the multiple possible solutions of this problem. Next, for searching the unique solution Darwin's theory of resolution establish this under the bound of "Survival of the fittest". The best solution is selected and the rest is eliminated. This process is repeated until the selection of "The Real Best." The advantage of this algorithm is that there is a spectrum of possibilities of the single problem of deciding to the desired. Hence, GA is a efficient searching algorithm to achieve unique and single solution among the infinite solutions.

Basically, GA is a coding technique by the finite bit strings. Its combination plays an important role to establish the system. Thus, it interacts with the mathematics. The strings of the bit represented as;

1	0	1	0	0	1	0	1	1	0

This selects as the elements of the coding with the following associate preliminaries:

4.2.1 Definition

Let a function $f : x \geq R \geq 0$, then there will be an optimization problems as, $x_0 = \arg\{\max f(x)\}$

Where $f(x)$ be a fitness function,

$x = (x_1, x_2, x_3, x_4 \ldots\ldots\ldots\ldots\ldots x_n)$ be a binary vector,

Or

$x_i \in \{0,1\}$

$X_i : variable$

$x_i : assignments$

The above definition interacts with the objective function, but this depends on the following crossover as the constraints.

4.2.2 Definition

Let *x*, *y* be the two offsprings, *z* be a one-point crossover under the interval $0 < \mathrm{i} < \mathrm{n}$ such that $z_i = x_i$ for $\mathrm{i} \leq 1$ and $z_i = y_i$ for $\mathrm{i} > 1$. The uniform crossover z_i is choosing randomly from $\{x_i, y_i\}$.

The next definition for its probability distribution is explained as follows:

4.2.3 Definition

At the generation *t*, $p(x,t)$ supposed as probability if *x* in the population $p_i(x_i,\ t) = \sum p(x_i,\ t)$ is the invariable marginal distribution.

The selection process proportionates to the followings:

4.2.4 Definition

The rate of selection of the probable reproduction is defined as follows:

$$p(x,t+1) = p(x,t)\frac{f(x)}{p(t)} \tag{4.1}$$

For the crossover or the recombination such different probability distribution is required, which is given below:

4.2.5 Definition

Let the probability distribution is defined by π as,

$$\pi(x,t)=\prod_{i=1}^{n}p_i(x_i,t) \tag{4.2}$$

This is called "the Robbins proportion."

If both the above characteristics, i.e. selection and crossover, are adjoined then its mathematical expression can be defined as follows:

$$p(x,t+1)=\sum R_{x,y,z}\,P^s(y)\,P^s(z) \tag{4.3}$$

Where, *R* is a crossover or recombination probability distribution, $R_{x,y,z}$ is the probability of recombination of x, y and z, $P^s(y)$ is the probability of strings over *y*, $P^s(z)$ is the probability of strings over *z*.

Next section presents the Mobile Robot Navigation based on the application of GA.

4.3 Proposed Matrix-Binary Codes based GA Controller

This section proposes the matrix based study of the GA for MRN controller. Here GA is presented as optimized tool for searching the best fit path for single and multiple MRN problems. The proposed controller uses the matrix trace based mechanism to sequence the operation during the navigation and intelligent GA searches the goal by avoiding obstacle. The searching process of environment is classified by two ways such as linear, nonlinear search. The current study using GA deals with the nonlinear search which processes iteratively. The process begins with the input as LOD, FOD and ROD from the sensors to output as a desired heading angle (HA). The iteration process corrects the output upto the marks of optimization. The proposed method is structured between the start points of navigation to goal point of navigation for general and complex type environment. By the proposed controller, the real-time mobile robots decision, navigation, generalization, optimization and correction are presented in this chapter.

GA is the function of gene's chromosomes and genetic operators thus the efficiency is proportional to these variables. By the restructuring these variables the GA can be performed better than the earlier. Basically, GA is a genetic type structure by its genetic operators and structure. The sequencing, ordering, selecting and grouping the variables can be redefined. These new genetic structures may become the foundation for a further effective step towards the best-fit decision. The population of chromosomes is used as the operand and GA as the operator. The infrared sensors measures the distances between the robot and obstacles and give the inputs as FOD, ROD, and LOD to controller and Matrix trace mechanism finds the best fit heading angle (HA) as output. There are the crisp sets which are encoded as the discrete frequency distribution. Let the crisp sets are transformed into binary sets. It is the input data code which will be given to robot controller via the sensors mechanism. Although each set defined and encoded but its transformation is according to the following layers:

Layer 1: The combination set

Layer 2: The fitness function

Layer 3: The crossover

Layer 4: The mutation

Layer 5: Evaluation of fittest child

The behavior of all the layers mentioned above is presented below:

Layer 1: The Combination Set: - Let the population set be $P = \{P_1, P_2 .... P_n\}$

The number of chromosome in $P_i = 5$

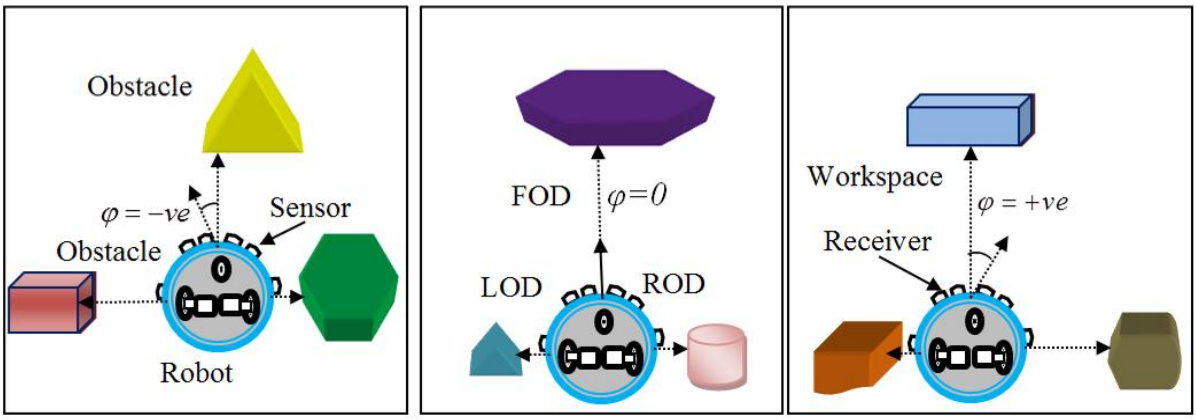
The structure of the element $= (i, j)$

Then, the matrix representation will be,

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} & P_{15} \\ P_{21} & P_{22} & P_{23} & P_{24} & P_{25} \\ P_{31} & P_{32} & P_{33} & P_{34} & P_{35} \\ P_{41} & P_{42} & P_{43} & P_{44} & P_{45} \\ P_{51} & P_{52} & P_{53} & P_{54} & P_{55} \\ . & . & . & . & . \\ . & . & . & . & . \\ P_{n1} & P_{n2} & P_{n3} & P_{n4} & P_{n5} \end{bmatrix} \tag{4.4}$$

Each column of the matrix P is represented the individual characteristics of the GA structure and it is given in the following:

$$FOD = \begin{bmatrix} P_{11} \\ P_{21} \\ P_{31} \\ P_{41} \\ P_{51} \\ . \\ . \\ P_{n1} \end{bmatrix}, LOD = \begin{bmatrix} P_{12} \\ P_{22} \\ P_{32} \\ P_{42} \\ P_{52} \\ . \\ . \\ P_{n2} \end{bmatrix}, ROD = \begin{bmatrix} P_{13} \\ P_{23} \\ P_{33} \\ P_{43} \\ P_{53} \\ . \\ . \\ P_{n3} \end{bmatrix}, HA = \begin{bmatrix} P_{14} \\ P_{24} \\ P_{34} \\ P_{44} \\ P_{54} \\ . \\ . \\ P_{n4} \end{bmatrix} \;\&\; Signconversion(\mathrm{SC}) = \begin{bmatrix} P_{15} \\ P_{25} \\ P_{35} \\ P_{45} \\ P_{55} \\ . \\ . \\ P_{n5} \end{bmatrix} \tag{4.5}$$



Scenario 1 (Case: close right obstacle) Scenario 2 (Case: obstacle far around) Scenario 3 (Case: close left obstacle)

Figure 4.1: Output of the GA regarding HA

Table 4.1: Heading angle of the robot as per the distance from the obstacle

Case No.	LOD in cm	ROD in cm	FOD in cm	HA (φ) in degree	Direction
1	10	12	60	0	Straight
2	45	18	32	9	Negative
3	10	45	50	15	Positive
4	20	36	12	11	Positive
5	28	16	27	14	Negative
6	13	15	80	0	Straight
7	34	27	380	12	Negative

Layer 2: The Fitness Function: The evolutionary process depends on the fitness function. The generation and justification of the fitness function are essential for the execution of the navigational controller for optimal decision. The efficiency of the obstacle avoidance mechanism and optimal path generation depend on the accuracy of the fitness function. Hence, the fitness function is proportional to the obstacle avoidance and path optimization. The controller follows the optimal function, which is proposed as follows,

$$f_{Total} = (c_1, c_2, c_3, c_4, c_5) T_r(p) \tag{4.6}$$

Here $T_r(p)$ is the Trace, i.e. the sum of the diagonal elements of p

$$T_r(\mathrm{p}) = \begin{bmatrix} f_1 & . & . & . & . \\ . & f_2 & . & . & . \\ . & . & f_3 & . & . \\ . & . & . & f_4 & . \\ . & . & . & . & f_5 \end{bmatrix} \tag{4.7}$$

$$\text{Where, } f_1 = \begin{bmatrix} \left(c_{FOD} - p_{ci,1}\right)^2 & . & . \\ . & \left(c_{FOD} - p_{ci,1}\right)^2 & . \\ . & . & \left(c_{FOD} - p_{ci,1}\right)^2 \end{bmatrix} \tag{4.8}$$

$$f_2 = \left|\left(c_{FOD} - p_{ci,1}\right)\right| \tag{4.9}$$

$$f_3 = \left|\left(c_{LOD} - p_{ci,2}\right)\right| \tag{4.10}$$

$$f_4 = \left|\left(c_{ROD} - p_{ci,3}\right)\right| \tag{4.11}$$

$$f_5 = \left|TA - HA\right| \tag{4.12}$$

The best optimum child be $\left(c_{FOD} - p_{ci,1}\right)$, $\left(c_{LOD} - p_{ci,2}\right)$ and $\left(c_{ROD} - p_{ci,3}\right)$.

Here, the Target Angle is TA and Heading Direction is HA.

Layer 3: The Crossover: The crossover is studied over the probability distribution. In this process, the chromosomes are selected initially from the chosen parents and then it is an applied for crossover according to probability. The two parent chromosomes used as the generator for one-point crossover. The gene data set is also required to produce the two offspring chromosomes for the next crossover operation. In proposed controller, the heading angle is the decision variable for getting the new location by obtained rule. The Matrix-Binary Code generation is processed as below:

The binary set $\{0,1\}$ and its representation for the linguistic variables by the distribution, combination and tabulation, are presented below:

Table 4.2: Logic decision table:

S.No.	Linguistic Variable	Matrix Representation
1	Very Far	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
2	Far	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$
3	Close	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
4	Near	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
5	Very Near	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
6	Very Fast	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$
7	Fast	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$
8	Medium	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$
9	Slow	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$
10	Very Slow	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$

Crossover for F.O.D.: It is represented by the product of the Matrixes of parents I & II and the offspring 1 & 2:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \\ b_{51} & b_{52} \end{bmatrix} \tag{4.13}$$

Similarly the crossover for LOD:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \\ b_{51} & b_{52} \end{bmatrix} \tag{4.14}$$

Similarly the crossover for ROD:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \\ b_{51} & b_{52} \end{bmatrix} \tag{4.15}$$

And similarly for the HA:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \\ b_{51} & b_{52} \end{bmatrix} \tag{4.16}$$

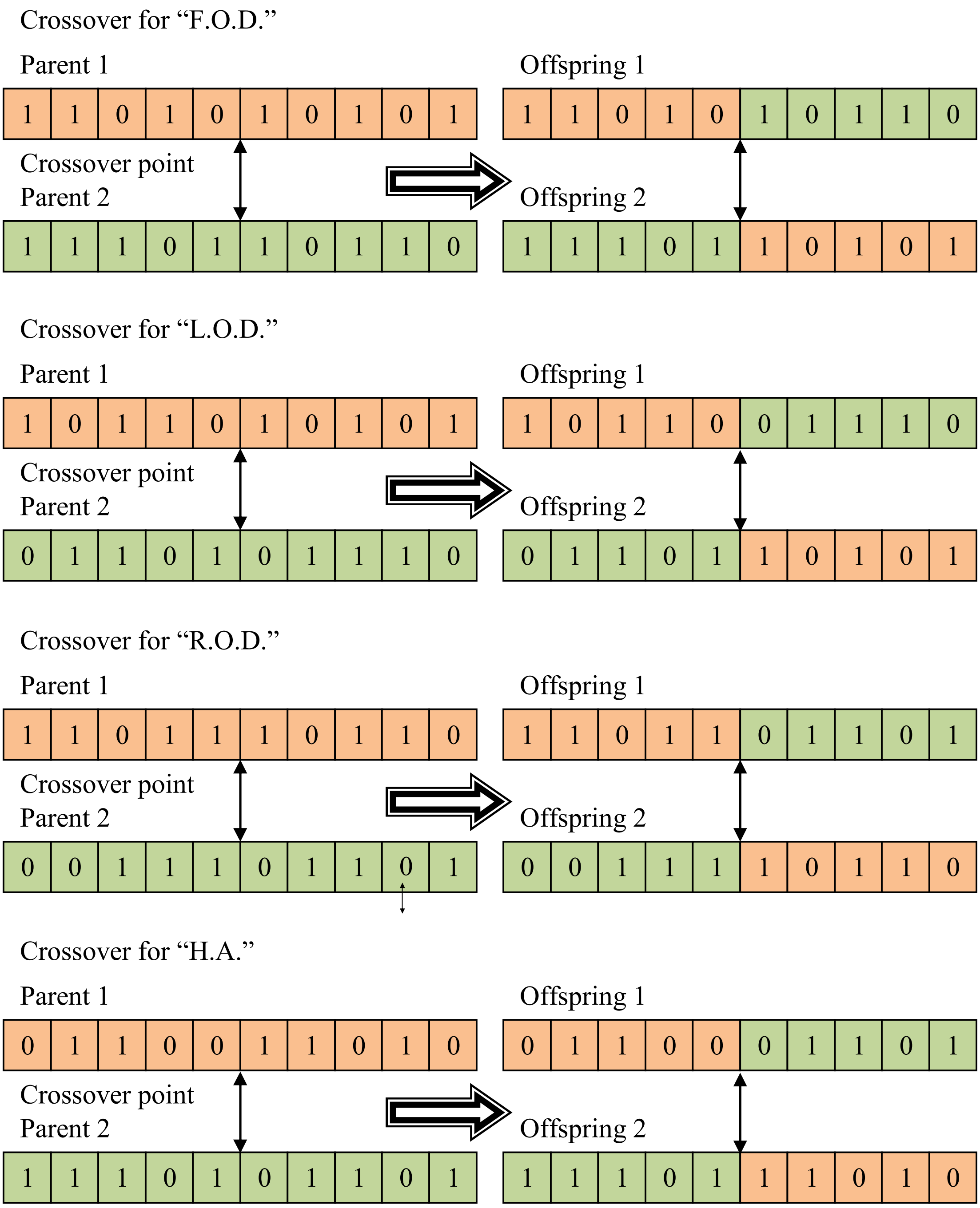


Figure 4.2: Simple crossover mechanisms

Layer 4: Mutation: The combination of genes is responsible for every genetic operation. The rate of change of chromosomes is measured and referred by mutation operator. The mutation operator generates the obstacle avoidance behavior. The sequence correction with the one to one correspondence is termed in mutation operation. The procedure of mutation is described as a random number, population, direction and distribution.

The following is given as proposed mutation technique:

Let, the mutation vector set be $\lambda = \{\lambda_1, \lambda_2, ...\lambda_n\}$ and the set of chromosomes be

$c = \{c_1, c_2, ...c_n\}$. There will be a certain rule *f*, which refers as the mutation operator. Let $f(\lambda)$ is a mutation operator, which is the range of the function from λ to *c*.

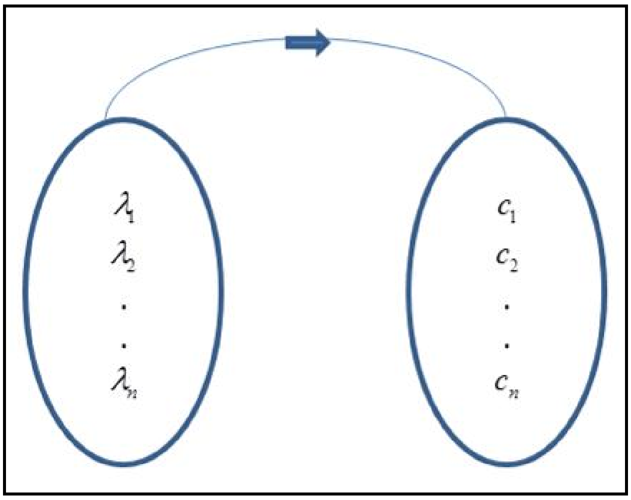


Figure 4.3: Mutation operator

While analyzing, the domain is λ; co-domain is *c* and the range of *f* is $f(\lambda)$. But the generation point of view, these should be a postulate, to execute the operation under the natural behavior of genes. For the mutation operation, the location of the chromosomes can be represented by the following matrix:

$$c_L = \begin{bmatrix} c_{11} & c_{12} & . & . & c_{1n} \\ . & . & . & . & . \\ . & . & . & . & . \\ c_{m1} & c_{m2} & . & . & c_{mn} \end{bmatrix} \tag{4.17}$$

Hence, the mutation operation is applying as,

$$\lambda_1 \begin{bmatrix} c_{11} \\ . \\ . \\ c_{m1} \end{bmatrix} + \lambda_2 \begin{bmatrix} c_{12} \\ . \\ . \\ c_{m2} \end{bmatrix} + ...+ \lambda_n \begin{bmatrix} c_{1n} \\ . \\ . \\ c_{mn} \end{bmatrix} = \begin{bmatrix} \lambda_1 c_{11} \\ . \\ . \\ \lambda_1 c_{m1} \end{bmatrix} + \begin{bmatrix} \lambda_2 c_{12} \\ . \\ . \\ \lambda_2 c_{m2} \end{bmatrix} + ... \begin{bmatrix} \lambda_n c_{1n} \\ . \\ . \\ \lambda_n c_{mn} \end{bmatrix}$$

$$= \begin{bmatrix} \lambda_1 c_{11} + \lambda_2 c_{12} + ...+ \lambda_n c_{1n} \\ . \\ . \\ \lambda_1 c_{m1} + \lambda_2 c_{m1} + ...+ \lambda_n c_{mn} \end{bmatrix} = \begin{bmatrix} D_1 \\ . \\ . \\ D_n \end{bmatrix} \tag{4.18}$$

Thus, the mutation operator vector is,

$$D = \{D_1, D_2 \,...........\, D_n\} \tag{4.19}$$

Layer 5: Evaluation of Fittest Child: It is proportional to the fitness function. There exists an infinite set of possible solutions, but the requirement is unique. Hence, GA applies for searching this. There are the two classes of processing this objective. The first is by the computation and the second is probabilistic distribution. The computation process forms a subset of the existing randomized solutions and by the probabilistic distribution the elements of this subset forms sequentially. The key objective is "To get the best HA". Let it be φ, it requires to put in the particular algorithm, to establish φ and then the fixed mutation operation be executed. The proposed algorithm consist the rule *f*, which is defined in the previous section. Its range, domain and co-domain remain the same but according to the environment, the mutation operator matrix is changed. This nature lays the biological properties of chromosomes.

4.4 Simulation Analysis

The current section provides the capability of the proposed controller for MRN in different environments. The variety of the environment has been developed in Matlab simulation software for MRN to checking the effectiveness of the proposed controller regarding path length and required time of navigation. The Matlab (R2008) software helps to perform program successfully for single or multiple mobile robots, multiple targets and multiple goal in the prescribed boundary. The simulation analysis has been conducted on the PC with I3 processor (3GHz), 4 GB RAM, 500 GB hard disk, Windows 7 (64 bit) OS, NVDIA (1GB) graphics card. The simulation results have been tested in 2D space of a100cm by a 100cm square background in the presence of a variety of static and dynamic obstacle. While navigating in the environment robot follows the shortest path between the goal and robot. The Euclidian distance calculates the shortest path. When an obstacle appears in the path of the robot then the proposed Matrix based GA controller activates. The robot starts calculating the LOD, ROD and FOD. The GA-based controller finds the best-fit rule to get desired heading angle of navigation. The Figures (4.4-4.7) demonstrate the efficiency of the proposed navigational controller which effectively avoids the obstacle. The Figure shows the robot path from start point to goal point while negotiations with obstacles. The robot shows the wall following behavior during the navigation when the length of the walls is too long.

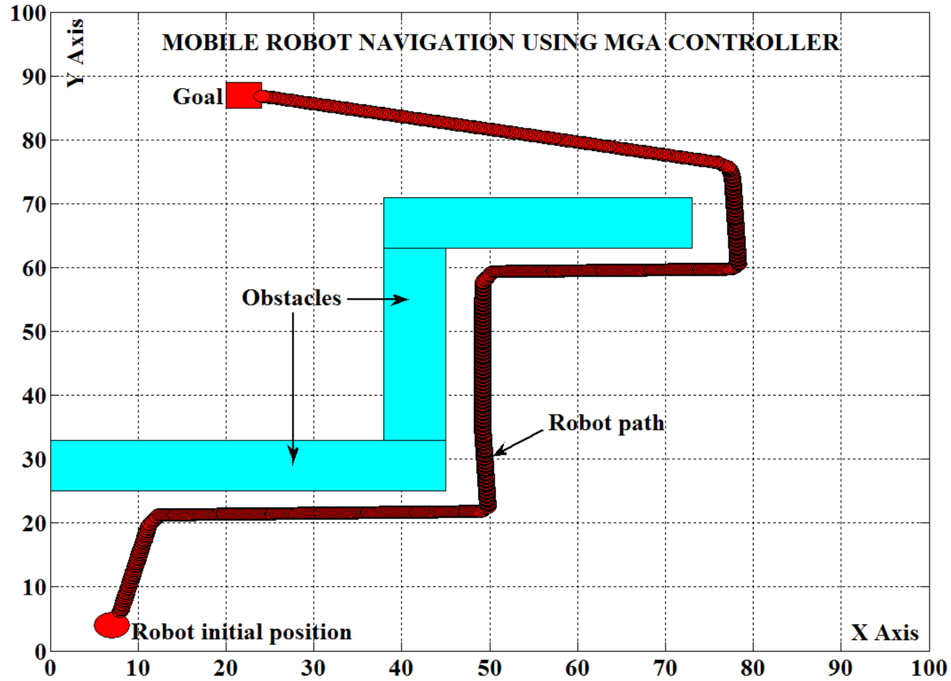


Figure 4.4: Navigation using MGA controller

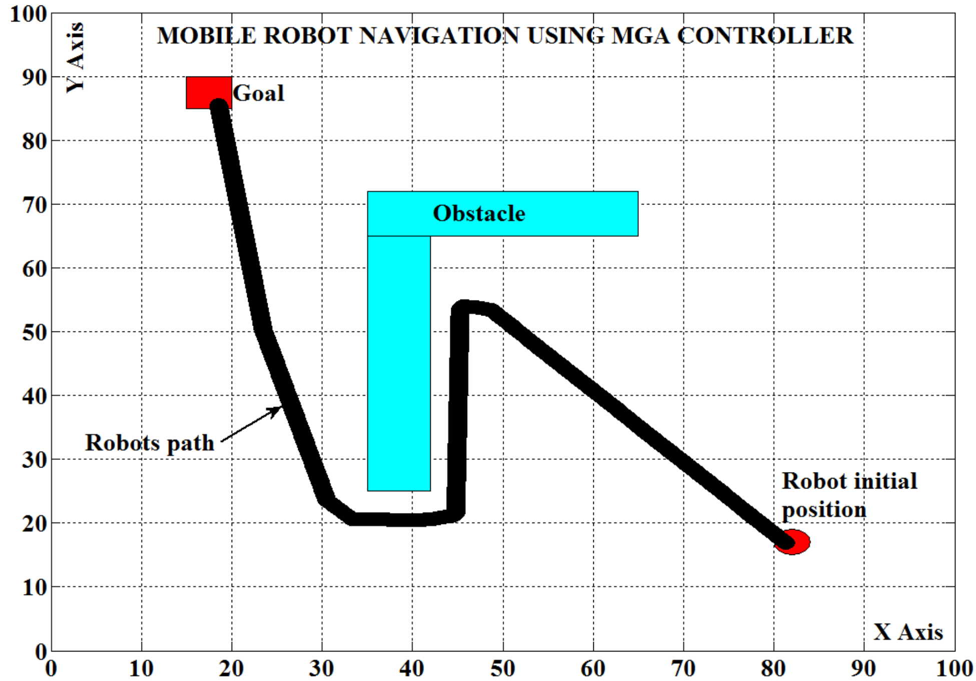


Figure 4.5: Navigation using MGA controller

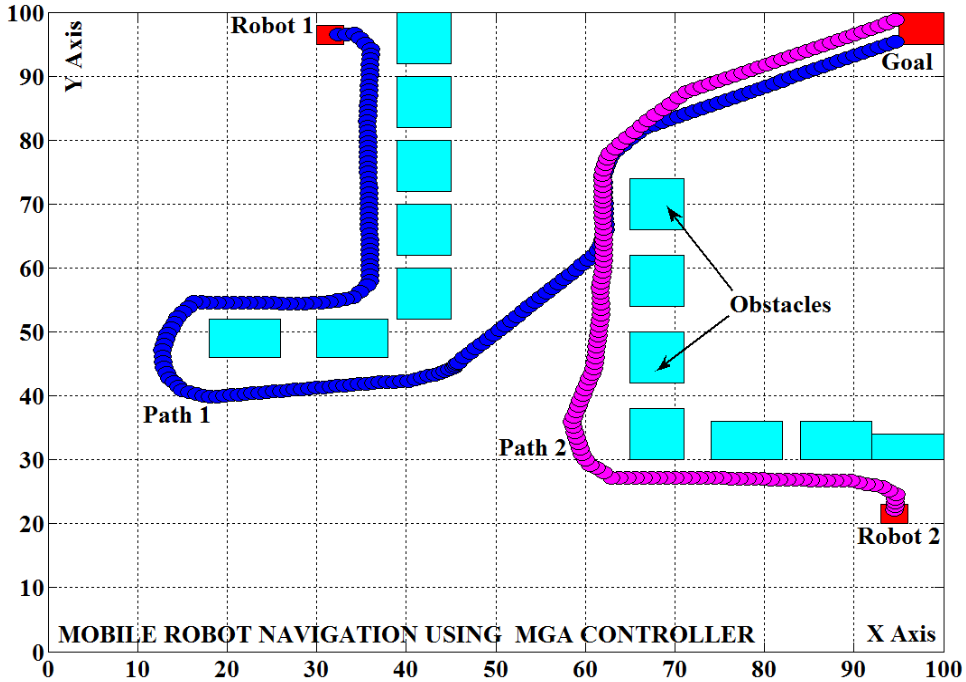


Figure 4.6: Navigation of multiple mobile robots using MGA controller

The simulation analysis is also carried out in the presence of the dynamic obstacle. The environment with three dynamic obstacles is shown in Figure 4.7 The robot one by one avoids the obstacle by making safe distance when obstacles come in the sensory range of the robot.

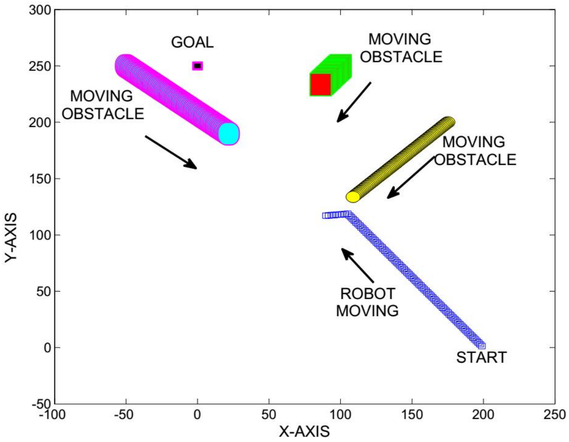


Figure 4.7 (a)

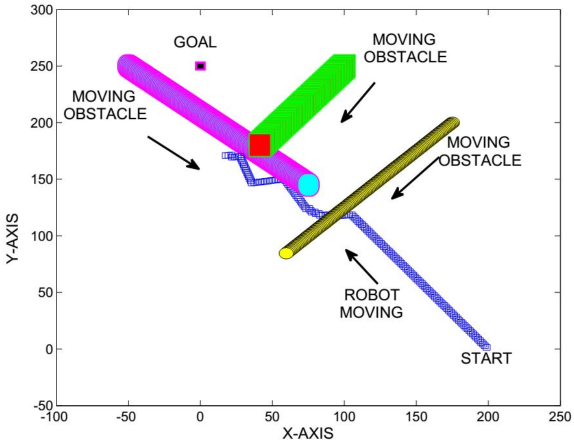


Figure 4.7 (c)

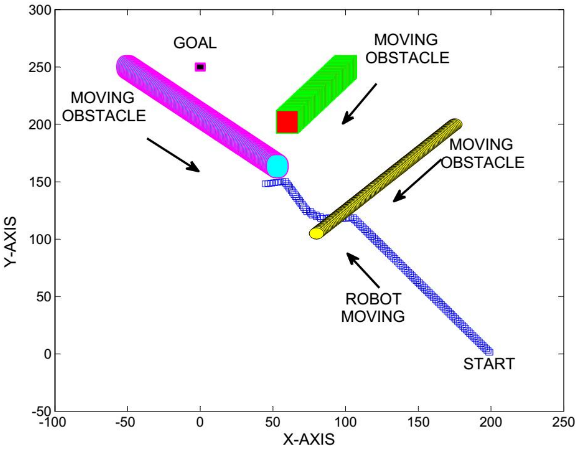


Figure 4.7 (b)

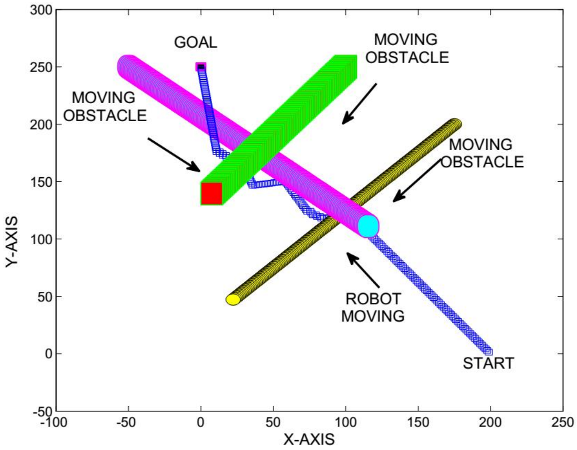


Figure 4.7 (d)

Figure 4.7: Navigation in presence of dynamic obstacles using MGA controller

4.5 Experimental Analysis

The experimental analysis is presented in this section for demonstrating the validity of the proposed controller for real time navigation. The environment with rectangular boundries is developed in the laboratory. While creating the environment, the robot platform is kept plain for smooth motion of robot. For experimental analysis, the Khepera-II robot is considered. The robot has eight infrared sensors arranged in a circular fashion to understand the environment. The other specification of the robot is mentioned in the Appendix section. The setup of the environment is kept in such a way that robot unable to see the goal due to the presence of the obstacle. The robot starts sensing the environment and traces the location of the goal and obstacles and follows the shortest possible distance for navigation. To avoid the obstacle present in the path, the robot is encoded with the MGA controller by using C++. The proposed controller starts working when robot detects the obstacles. The robot follows the evolutionary genetic based mechanism to avoid the obstacle and finds the best fit heading angle to achieve the goal. The Figures (4.8-4.10) show step by step path planning of mobile robot in the presence of a static obstacle. The presented experimental analysis of single and multiple mobile robots proves the path optimality as it generates the appropriate path between the robot and goal; it also avoids the random movement of the robot which leads to the lengthy path.

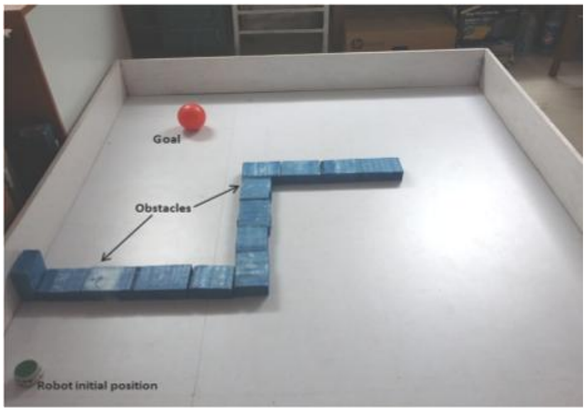


Figure 4.8 (a)

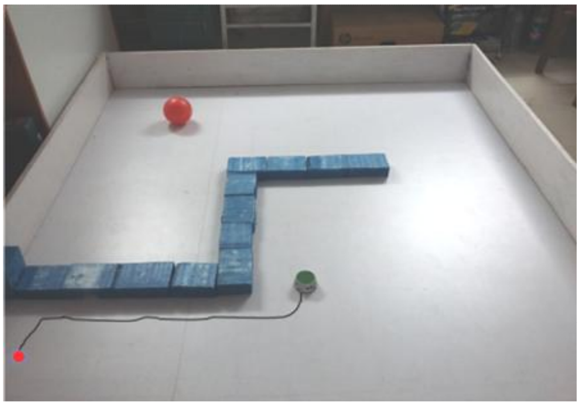


Figure 4.8 (b)

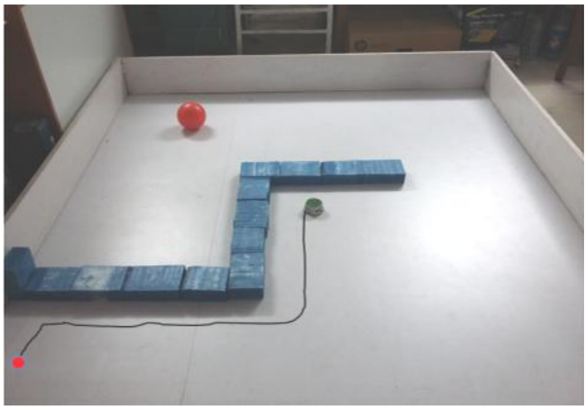


Figure 4.8 (c)

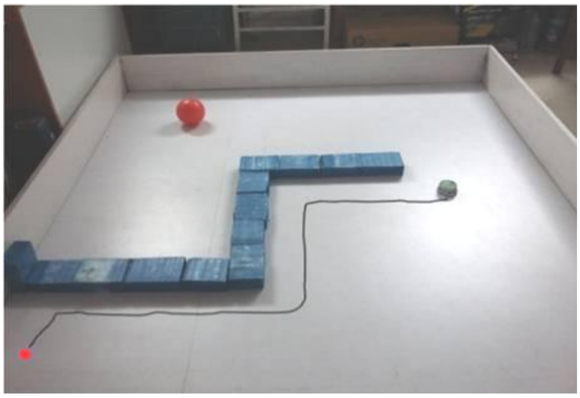


Figure 4.8 (d)

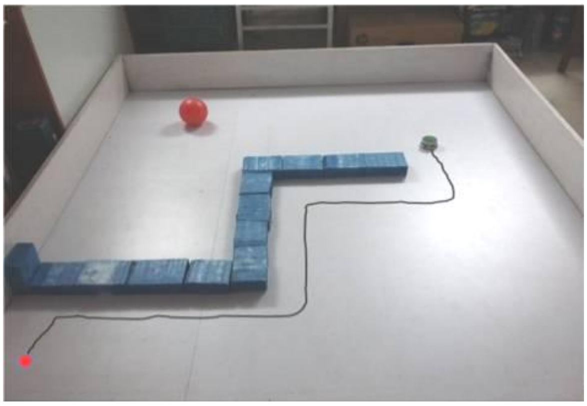


Figure 4.8 (e)

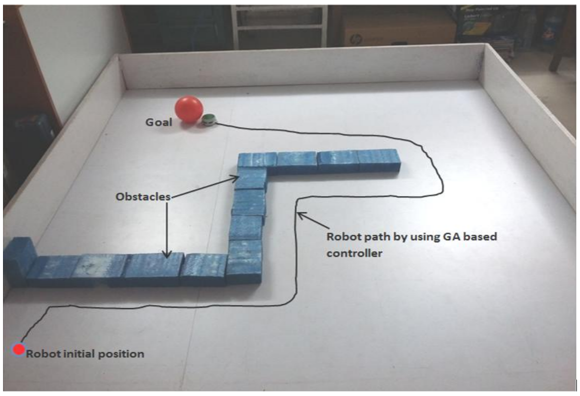


Figure 4.8 (f)

Figure 4.8: Real-time navigation using MGA controller

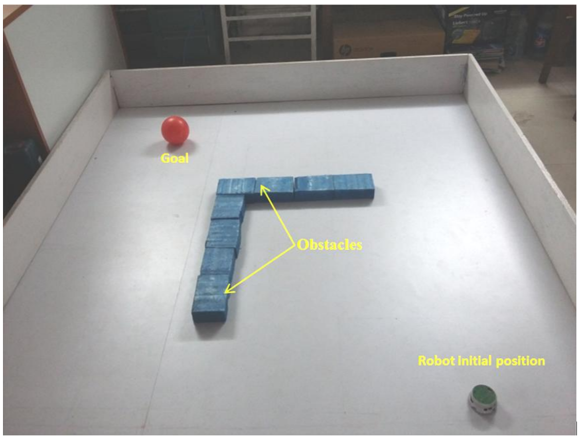


Figure 4.9 (a)

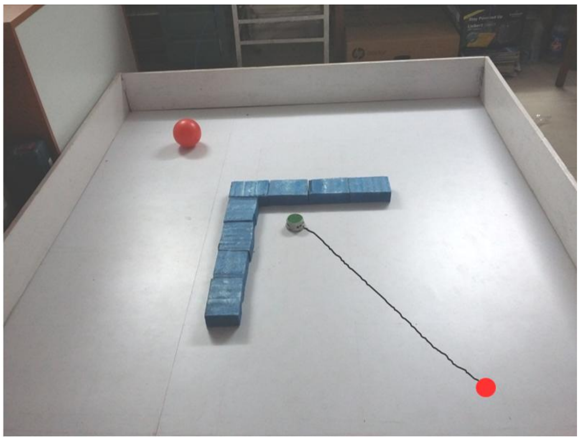


Figure 4.9 (b)

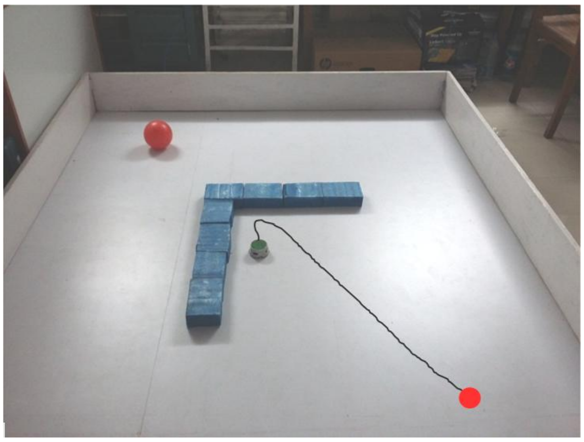


Figure 4.9 (c)

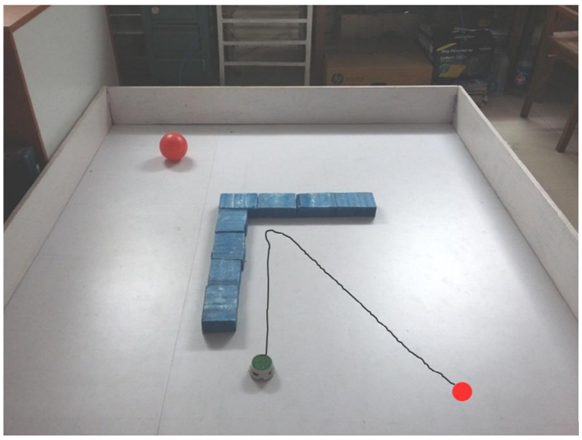


Figure 4.9 (d)

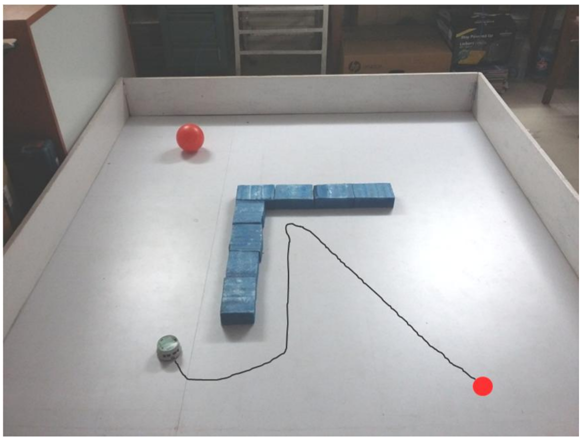


Figure 4.9 (e)

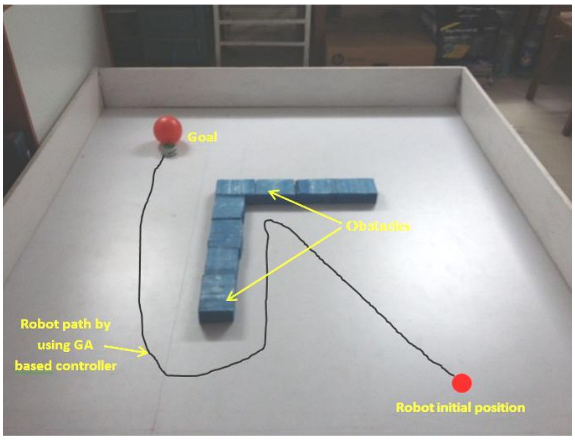


Figure 4.9 (f)

Figure 4.9: Real-time navigation using MGA controller

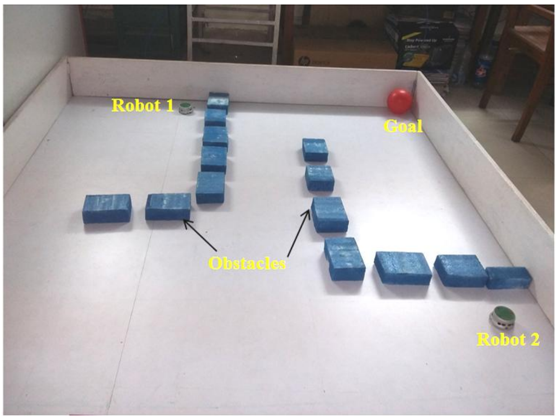


Figure 4.10 (a)

Figure 4.10 (b)

Figure 4.10 (c)

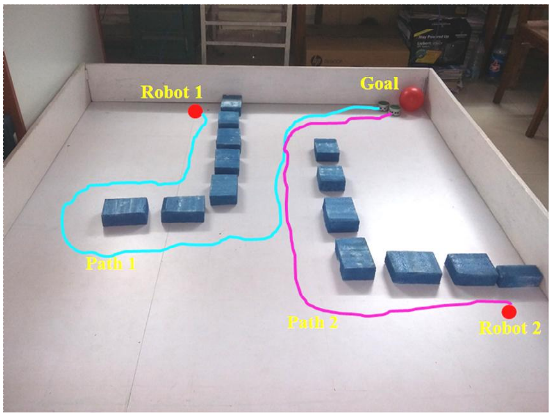


Figure 4.10 (d)

Figure 4.10: Real-time navigation for multiple mobile robots using MGA controller

4.6 Comparative Study of Experimental and Simulation Analysis of MRN over Similar Environment

In this section, the experimentally observed path length and time taken during navigation is compared with the simulational results. The comparison between the experimental and simulational analysis is carried over the two similar environmental setups for the single mobile robot and multiple robot systems. To verify the performance of the robot during the simulation and the real-time experiment, the many trials are taken to calculate the path length and time requires for navigation. The observed path in the simulation closely follows the walls. In Scenario -1, the Figures (4.4 and 4.8) are compared, in Scenario-2, the Figures (4.5 and 4.9) and in Sceneraio-3 the Figures (4.6 and 4.10) are compared to understand the performance. The observed path length and required time of navigation for 20 trials are tabulated in Tables (4.3-4.6) for single mobile robot where as 10 trials are taken for multiple mobile robots system as shown in Figure (4.7-4.8). The observed path length is minimum in case of simulational analysis as compared to experimental analysis for same environmental setup. The obtained percentage of deviation for path length is within 5.5% for single and multiple mobile robot system. The time required to accomplish the same task of navigation in the experimental analysis is more than the time required in simulational analysis. The observed percentage of deviation is less than 6% between the simulational and experimental results single and multiple robot system.

Table 4.3: Path length in same simulational and experimental setup (Figure 4.4 and 4.8).

No. of runs	Experimental path length during MRN (in'cm')	Simulational path length during MRN (in 'cm')	% of deviation
1	338.2	321.9	4.81
2	337.07	321	4.76
3	337.61	320.77	4.98
4	342.09	324.05	5.27
5	339.19	324	4.47
6	337.89	323.31	4.31
7	338.09	323.33	4.36
8	339.4	323.68	4.63
9	338.95	321.76	5.07
10	338.59	321.5	5.04
11	339.55	320.81	5.51
12	340.54	320.42	5.90
13	339.22	323.3	4.69
14	340.93	321.11	5.81
15	341.58	321.98	5.73
16	342.94	320.8	6.45
17	343.87	321.76	6.42
18	344.85	323.8	6.10
19	345.5	323.57	6.34
20	346.21	323.05	6.68
Average path length covered	**340.31**	**322.25**	**5.30**

Table 4.4: Path length in same simulational and experimental setup (Figure 4.5 and 4.9).

No. of runs	Experimental path length during MRN (in'cm')	Simulational path length during MRN (in 'cm')	% of deviation
1	313.03	296.1	5.40
2	314.52	295.89	5.92
3	313.41	297.87	4.95
4	313.25	296.15	5.45
5	314.94	297.12	5.65
6	311.71	296.53	4.86
7	313.52	298.21	4.88
8	312.1	298.19	4.45
9	313.93	295.49	5.87
10	314.14	297.44	5.31
11	315.97	297.05	5.98
12	316.56	297.87	5.90
13	311.19	297.48	4.40
14	312.18	296.67	4.96
15	311.98	297.11	4.76
16	311.63	295.66	5.12
17	311.39	296.54	4.76
18	313.97	297.15	5.35
19	315.75	298.15	5.57
20	316.21	298.25	5.67
Average path length covered	**313.43**	**296.98**	**5.24**

Table 4.5: Navigational time in same simulational and experimental setup (Figure 4.4 and 4.8).

No. of runs	Experimental time during MRN (in 'sec')	Simulational time during MRN (in 'sec')	% of deviation
1	18.6	17.65	5.10
2	18.69	17.72	5.18
3	18.82	17.7	5.95
4	19.2	17.94	6.56
5	18.5	17.69	4.37
6	19.35	18	6.97
7	18.74	17.84	4.80
8	19.77	18.2	7.94
9	19.95	18.68	6.36
10	18.94	18.22	3.80
11	18.97	18.3	3.53
12	18.41	17.48	5.05
13	19.83	18.87	4.84
14	18.98	18.3	3.58
15	18.67	17.91	4.07
16	20	18.76	6.2
17	19.85	18.75	5.54
18	18.61	18	3.27
19	20.1	18.79	6.51
20	20.1	18.7	6.96
Average time required	**19.20**	**18.17**	**5.33**

Table 4.6: Navigational time in same simulational and experimental setup (Figure 4.5 and 4.9).

No. of runs	Experimental time during MRN (in 'sec')	Simulational time during MRN (in 'sec')	% of deviation
1	16.9	16.1	4.73
2	16.5	15.6	5.45
3	17.15	16.46	4.02
4	16.9	15.4	8.87
5	16.6	15.3	7.83
6	17.7	16.8	5.08
7	17	16.35	3.82
8	17.33	16.75	3.34
9	16.62	15.8	4.93
10	16.5	15.37	6.84
11	16.54	15.2	8.10
12	17.6	16.31	7.32
13	18	16.88	6.22
14	18.2	17	6.59
15	16.75	15.5	7.46
16	16.75	16.1	3.88
17	17.55	17	3.13
18	17.31	16.2	6.41
19	17.95	16.9	5.84
20	18.2	17	6.59
Average time required	**17.20**	**16.20**	**5.82**

Table 4.7: Path length in same simulational and experimental setup (Figure 4.6 and 4.10).

No. of runs	Robot No.	Experimental path length during MRN (in'cm')	Simulational path length during MRN (in 'cm')	% of deviation
1	Robot 1	229.84	218.03	5.13
	Robot 2	189	179	5.29
2	Robot 1	229.6	217.1	5.44
	Robot 2	187.77	178.72	4.81
3	Robot 1	233.81	221.74	5.16
	Robot 2	189.25	181.72	3.97
4	Robot 1	230.94	218.5	5.38
	Robot 2	188.94	178.82	5.35
5	Robot 1	232	222.85	3.94
	Robot 2	190.1	180.7	4.94
6	Robot 1	232.7	221.8	4.68
	Robot 2	187.91	177.77	5.39
7	Robot 1	235.94	222.81	5.56
	Robot 2	190.8	179.12	6.12
8	Robot 1	234.22	222.45	5.02
	Robot 2	190.85	180.87	5.22
9	Robot 1	231.1	221.89	3.98
	Robot 2	190.56	180.25	5.41
10	Robot 1	234.7	223	4.98
	Robot 2	191	182.05	4.68
Average path length covered	**Robot 1**	**232.48**	**221.01**	**4.93**
	Robot 2	**189.61**	**179.90**	**5.12**

Table 4.8: Navigational time in same simulational and experimental setup (Figure 4.6 and 4.10).

No. of runs	Robot No.	Experimental time during MRN (in 'sec')	Simulational time during MRN (in 'sec')	% of deviation
1	Robot 1	37.05	35.4	4.45
	Robot 2	30.46	28.54	6.29
2	Robot 1	34.90	33.83	3.07
	Robot 2	31.68	29.80	5.91
3	Robot 1	38.01	35.59	6.37
	Robot 2	30.25	28.62	5.39
4	Robot 1	37.22	35.28	5.23
	Robot 2	32.40	31.03	4.24
5	Robot 1	35.69	34.41	3.58
	Robot 2	31.64	29.72	6.05
6	Robot 1	37.38	35.11	6.08
	Robot 2	30.02	28.65	4.54
7	Robot 1	34.51	32.65	5.39
	Robot 2	30.75	28.77	6.44
8	Robot 1	37.75	35.74	5.34
	Robot 2	29.76	28.85	3.05
9	Robot 1	35.25	33	6.39
	Robot 2	30.71	29.05	5.41
10	Robot 1	37.83	35.94	4.98
	Robot 2	31.78	30.34	4.53
Average time required	**Robot 1**	**36.56**	**34.69**	**5.09**
	Robot 2	**30.95**	**29.34**	**5.18**

4.7 Performance Analysis of MGA Controller with Other Navigational Controller

To prove the effectiveness of proposed controller, the comparative simulation analysis is presented with the other artificial intelligent controller. The Figure 4.11 & Figure 4.13 are the other navigational controller provided by Zhang [196] and Wang [197] respectively which is then compared with the developed MGA controller in Figures 4.12-4.14. The Zhang provided the hybrid neuro-fuzzy controller and Wang et al. presented the fuzzy logic controller for navigation of a mobile robot in a static environment. The similar environment is produced for comparison by using Matlab R2008. The data in Table 4.9 reflects that the proposed controller performs better than the existed navigational controller under the context of path optimality. The Maximum path length saved by proposed controller is up to 20%. The path is shown by proposed controller is very close to boundaries of obstacle hence, the proposed controller can be successfully used for robot navigation for the complex crowded environment in the presence of obstacles. The MGA based approach saves more path length and achieves the target within less time.

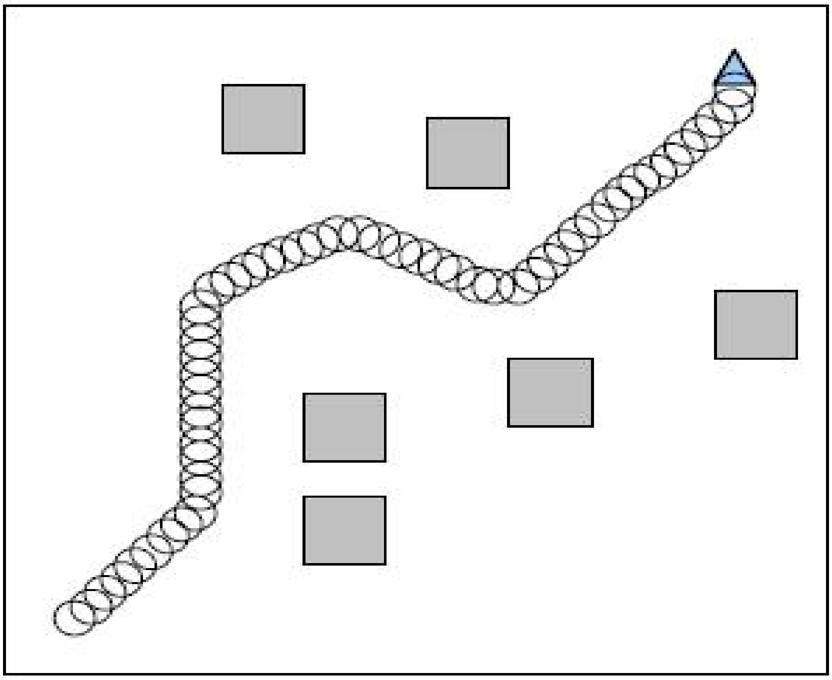


Figure 4.11: Navigation using neuro-fuzzy controller by Zang et al. [196]

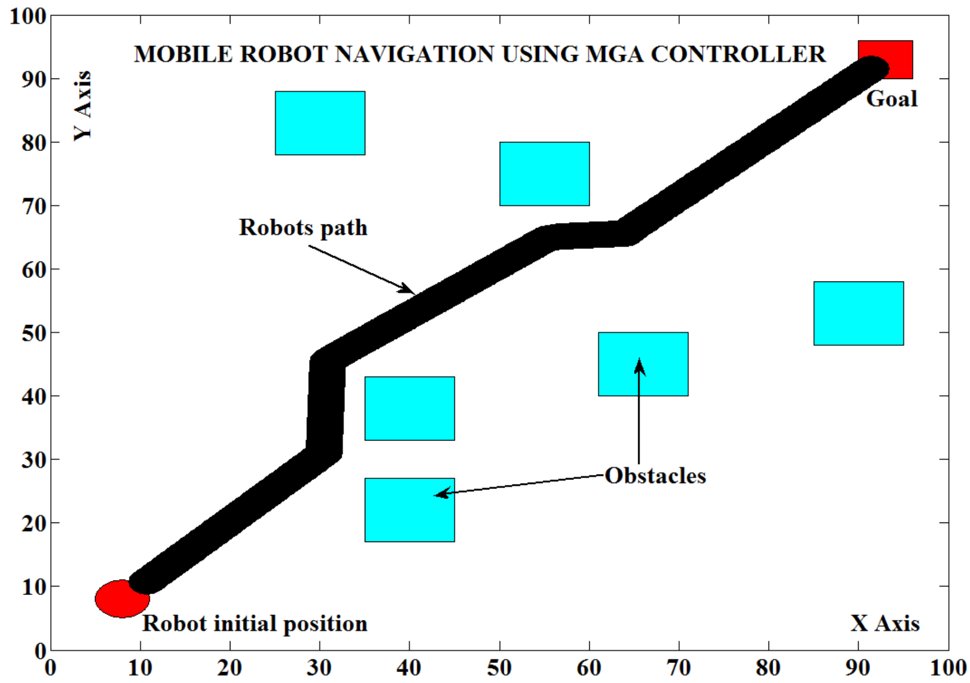


Figure 4.12: Navigation using MGA controller

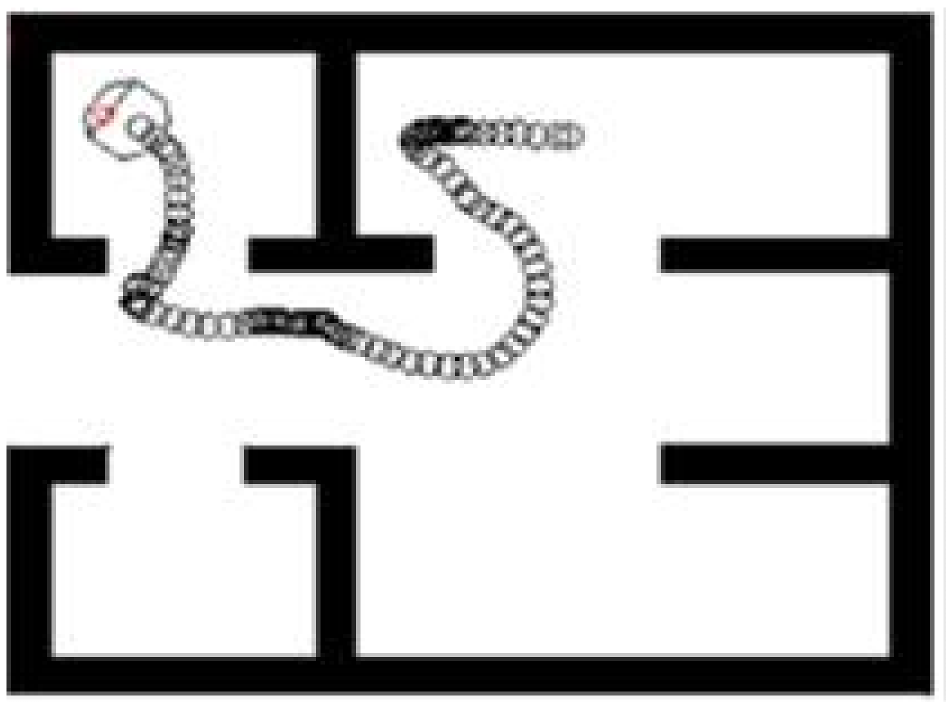


Figure 4.13: Navigation using fuzzy logic controller by Wang et al. [197]

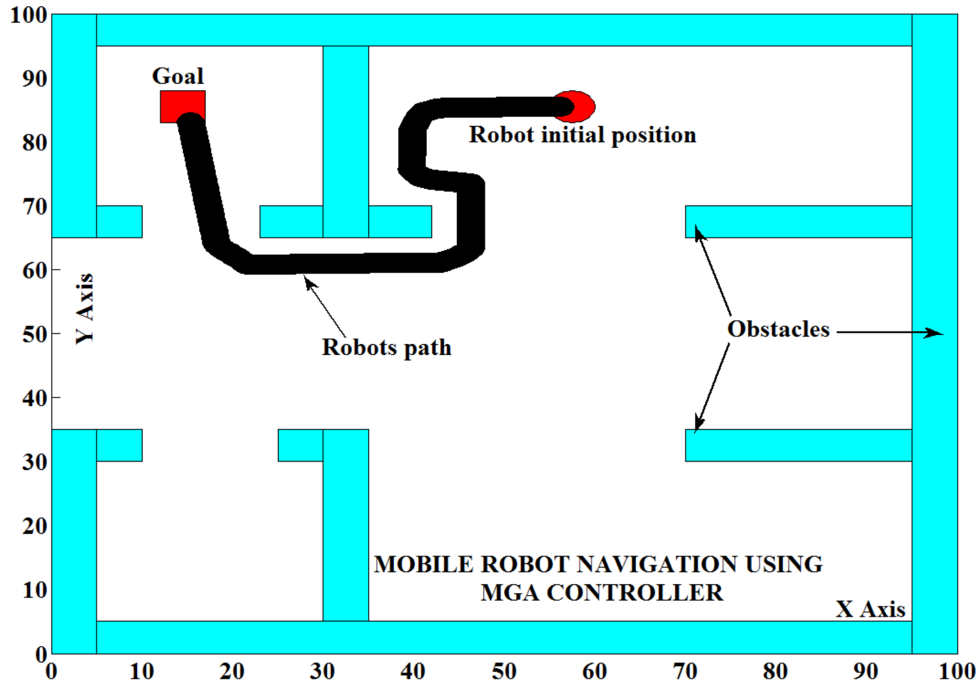


Figure 4.14: Navigation using MGA controller

Table 4.9: Comparision of simulation result regarding path length

Sl. No.	Simulational Path length (in 'cm') using MGA controller	Simulational path length (in 'cm') using other AI controller	% of path saved using MGA controller
Scenario-1	8.4 (Figure4.11)	9.8 (Figure 4.12)	14.28
Scenario-2	7 (Figure 4.13)	8.8 (Figure 4.14)	20

4.8 Summary

This chapter provides the application of the Matrix based Genetic algorithm for wheeled mobile robot navigation problem. The new MGA based approach is developed for robots to take their decision when working in uncertain environment. The key points of the finding are discussed as follows:

- The matrix trace based arrangement transforms GA into small sample space from the large sample space.
- The proposed controller finds the best string representation of optimum class of input or output and it sequences the solutions at the level of high flexibility.

- The controller distinguishes the discrete and continuous solution by the stochastic process. Robot requires this for generating the best fitness function.
- The obstacle avoidance behavior is more efficient for single and multiple mobile robots from the observed results.
- The proposed controller successfully avoids obstacles (static and dynamic) and also achieves the goal efficiently. The results of the navigation of mobile robot in presence of three moving obstacle is remarkable.
- The percentage of error observed for same experimental and simulational setup is less than 5.5% when compared for path length and it is observed less than 6% when compared for the time taken during navigation.
- On simultaneous comparison with other AI based controller (neuro-fuzzy and fuzzy logic algorithm), it is observed that proposed controller saves the path length upto 20%.
- At last, it is concluded that the proposed MGA controller can be successfully implemented for path planning problem in the uncertain environment.

Chapter 5

Probability-Fuzzy Logic Based Mobile Robot Navigation

This chapter presents a Mobile Robot Navigational controller based on Probability-Fuzzy logic function. To obtain optimal path between robot and goal in the presence of obstacles is the objective of the proposed work. But, it is challenging in the uncertain environment. Thus, the probability-fuzzy logic function is used for avoiding the obstacles and planning the optimal path. The central idea of this chapter is to form a function of probability and fuzzy logic over the distance and speed.

5.1. Introduction

The autonomous navigation of mobile robot involves various hidden challenges. The obstacle avoidance and optimal path decision is selected as the noteworthy problem in autonomous navigation. The probability-fuzzy logic based approach are introduced here to solve the navigational problem. Fuzzy logic interacts with obstacle avoidance & the optimal path decision with probability. In the current investigation fuzzy logic adjoins with probability by the common range [0, 1]. The chapter proposes a function whose domain is fuzzy set; co-domain is probability and distance-speed combination rule is the range. This new feature achieves both the said goals by the distance-speed combination. Multiple targets in known and unknown environments treating as the elements of the set and the decision require the fuzzy logic. The heuristics knowledge generates the fuzzy rule and its discrete distribution concerns with the probability. Robot path planning module is organized for mobile agents to map in a dense environment. It requires a justified strategy to execute an intelligent controller for the level of satisfaction over sensory information and reasoning barrier. The overview of fuzzy based mobile robot navigation is presented in accordance to the probabilistic theory.

5.2. Overview and Pre-requisites

The probabilistic theory is initiated by [0, 1] and the same range applies for fuzzy logic also. To establish an intelligent controller for a mobile robot, it requires the common membership function and this is obtained by probability-fuzzy logic common range i.e. [0, 1]. The vagueness, impression, lack of information, partial truth, uncertainties and oscillation interacts with fuzzy and probability distribution. This ordered pair is represented as the system of discrete probability distribution and continuous fuzzy membership function. The system generates the linguistic interpretation by the implicit knowledge. The fuzzy rule inter-relates the robotics decision and linguistic variable. The complex controlled task co-related with fuzzy, but it is a tedious and unreliable trial with errors.

The Pre-requisites are given below:

5.2.1 Definition

Consider a robotics fuzzy space and Let *X={O}* be a collection of obstacles in the robots path, then a robotics fuzzy subset *A* in *X* is a set of order pairs $A = \{(\mathrm{O}, \mu_A(\mathrm{O})\}$, where $\mu_A : O \rightarrow [0,1]$ is called the robotics fuzzy membership function of *A*.

5.2.2 Definition

The goal *G* in *X* is a subset of *X* characterized by its membership function $\mu_G(o)$.

5.2.3 Definition

Let *C* be a set of constraint in *X* is a subset of *X* defined by its membership function $\mu_C(o)$. Let, an angle measured by robot with the target be θ and its discrete probability distribution be;

θ	θ	.	.	.	θ
$p(\theta)$	$p_1(\theta\ )$	.	.	.	$p_n(\theta)$

The Discrete Fuzzy Constraint Distribution of the above is;

θ	θ	.	.	.	θ
$\mu_c(o)$	$(\theta, \mu_{c_1}(\theta)$	.	.	.	$(\theta, \mu_{c_n}(\theta)$

The composition of Probability-Fuzzy-Constraint Distribution is;

$p(\theta)$	$p_1(\theta)$	.	.	.	$p_n(\theta)$
$\mu_c(o)$	$(p_1(\theta), \mu_{c_1}(p_1(\theta))$	.	.	.	$(p_n(\theta), \mu_{c_n}(p_n(\theta))$

Similarly, The Discrete Fuzzy Goal Distribution of the above is;

θ	θ	.	.	.	θ
$\mu_G(o)$	$(\theta, \mu_{G_1}(\theta)$	.	.	.	$(\theta, \mu_{G_n}(\theta)$

The composition of Probability-Fuzzy-Goal Distribution is;

$p(\theta)$	$p_1(\theta)$	.	.	.	$p_n(\theta)$
$\mu_G(o)$	$(p_1(\theta), \mu_{G_1}(p_1(\theta))$	.	.	.	$(p_n(\theta), \mu_{G_n}(p_n(\theta))$

5.2.4 Definition

A robotics fuzzy decision D in X is obtained by the combination of G and C i.e $D = G \cap C$ and corresponding the membership function μ_D is defined by,

$$\mu_D(o) = \mu_G(o) \wedge \mu_C(o) = \min(\mu_G(o), \mu_C(o)) \tag{5.1}$$

$$or, D = G_1 \cap \ldots \cap G_n \cap C_1 \cap \ldots \cap C_n,$$

$$or, \mu_{G_1} \wedge \ldots \wedge \mu_{G_n} \wedge \mu_{C_1} \wedge \ldots \wedge \mu_{C_n}.$$

5.2.5 Definition

Let K be a fuzzy robotics Subset of X on which μ_D set as maximum, if it exists. An optimal decision is generally a robotics non-fuzzy subnormal subset D^* of D defined by,

$$\mu_{D^*}(o) = \begin{cases} \max \mu_D(o), o \in K \\ 0, elsewhere \end{cases} \tag{5.2}$$

5.2.6 Definition

Any X in the support of D^*, i.e. any alternative $o \in X$ which minimizing $\mu_D(o)$ is called the minimizing decision and denoted by o^*, thus,

$$\mu_D(o^*) = \min(\mu_{G_1}(o) \wedge \ldots \wedge \mu_{G_n}(o) \wedge \mu_{C_1}(o) \wedge \ldots \wedge \mu_{C_n}(o)) \tag{5.3}$$

5.2.7 Definition

If $f : A \rightarrow A$, then f is said to be a transformation function.

The probability-fuzzy logic function is the proposed technique of MRN in which the domain, co-domain and range of this function are formulated as below:

5.3. Problem Formulation

Mobile robots navigation and its operation are still challenging task in the world of artificial intelligence to decide the function from natural intelligence to artificial intelligence. This problem can be studied dynamically as a real-time problem, predefined goal problem, collision problem, trajectory problem, etc. The steering angle operates by said sequence as a distance-speed matrix. The heuristic knowledge about robots, target and obstacles fit in the ordered pair by probability-fuzzy logic rule. This is presented as an array of ultrasonic sensors, infrared sensors and collision sensors for detecting the bearing of the target. The controller comprises with real and virtual sensors as embedded MRN. The proposed model co-relates the position of obstacles, the movement of wheels, the direction of steering angle and the velocities of wheels. These are presented as the distance-speed matrix over the metric space [0, 1]. The inter-robot collision in Gaussian space for setting the membership function over Probabilistic-Fuzzy logic function transformed into the turning function. It lies to the obstacle avoidance in Gaussian space, which is represented in this model as discrete probability distributions. The inter-robot collision is represented as a fuzzy logic controller in the common range [0, 1]. Thus, probability-fuzzy logic controller behaves as the membership function of the mobile robot navigation controller. The experimental performance is also presented in this chapter to prove the proposed Probability-Fuzzy logic controller like a real and practical system. The distance-speed function is distributed as discrete and continuous pattern in Gaussian space over [0, 1]. It is presented as an objective function for optimizing the path under the subject to the obstacle avoidance over probability-fuzzy constraints.

5.4. Behavior-based Study of Navigation

Robot moves through measuring angle when navigating in its environment. There can be existed the two probable cases; this is analyzed below.

5.4.1 Case I (Without Obstacle)

Let R be a robot, G be a goal. The robot (R) identifies the angle with the goal (G) by the sensory information i.e. $\angle XRG = \theta$ as shown in Figure 5.1. Next, R allows to move towards G and $R \cong G$ or R and G are coincidence or $G \cong R$, thus $R = G$. It means R reached in G. This is the main objective.

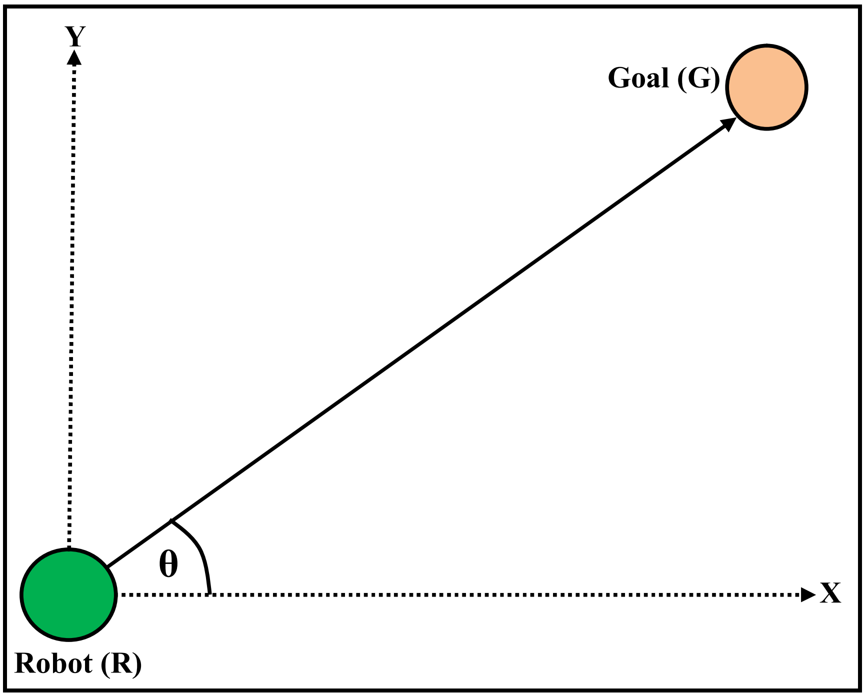


Figure 5.1: Robot environment without obstacle

The next case concerns with the presence of Obstacle in robots path.

5.4.2 Case II (With Obstacle)

Robot measures an angle θ, but there is an obstacle o in the path of robot, and if there are several obstacles then their sets can be represented as $O = \{o_1, \ldots, o_n\}$. Let, robot identifies the single obstacle say o_1 as shown in Figure 5.2 and starts to execute the obstacle avoidance process as given below:

1. Discrete probability distribution of θ.
2. Probabilistic-fuzzy logic rule Application.

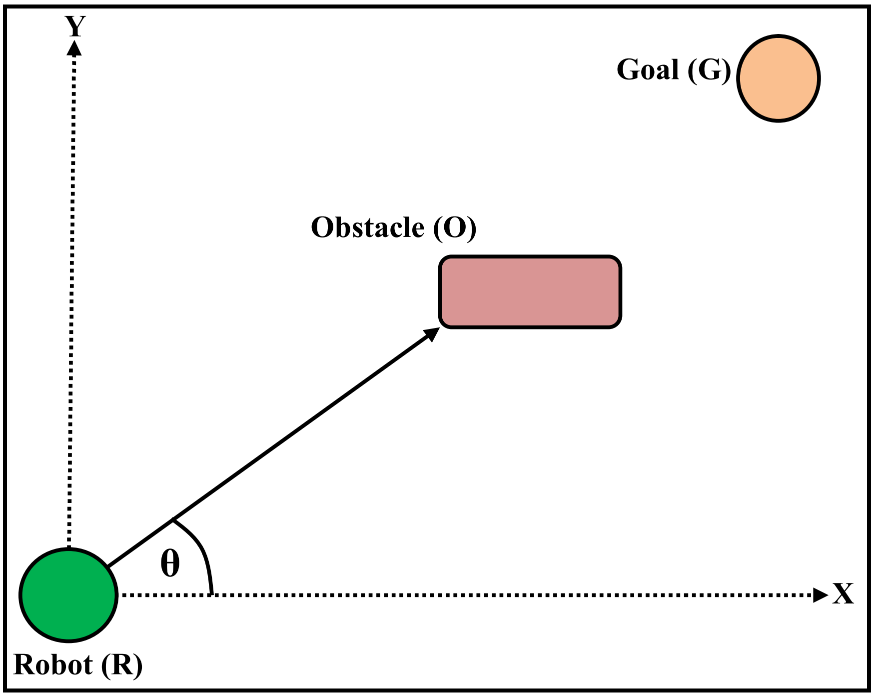


Figure 5.2: Robot environment with obstacle

Firstly, the angle θ is distributed discretely by the probability as per the tabulation mentioned above. But the second, which is the rule of fuzzy logic, is represented as below: Here, the three positions of an obstacle in respect of robot such as at left, at right and front side of the robot are considered during mobile robot navigation. The distance is measured from these three positions of the obstacles as Left Obstacle Distance (LOD), Front Obstacle Distance (FOD) and Right Obstacle Distance (ROD). The fuzzy rule with probability is presented in the following table:

Table 5.1: Probability-fuzzy logic rule

Obstacle Position	Distance Classification					Probability of Distance Classification					Combination of the Probability of Distance Classification
	D_1	.	.	.	D_n	$p(D_1)$	.	.	.	$p(D_n)$	$C_i(p(D_1),\ldots,p(D_n))$
LOD	d_{11}	.	.	.	d_{1n}	$p(d_{11})$	.	.	.	$p(d_{1n})$	C_1
FOD	d_{21}	.	.	.	d_{2n}	$p(d_{21})$	.	.	.	$p(d_{2n})$	C_2
ROD	d_{31}	.	.		d_{3n}	$p(d_{31})$	.	.		$p(d_{3n})$	C_3

It is generalized with the linguistic variables in the following tables:

Table 5.2: Probability-fuzzy logic rule with linguistic variable

Obstacle Position	Distance Classification						Probability of Distance Classification					Combination of the Probability of Distance Classification
	D_1	.	.	.	.	D_n	$p(D_1)$	.	.	.	$p(D_n)$	C_i
LOD	VVN	VN	N	F	VF	VVF	$p(d_{11})$	.	.	.	$p(d_{1n})$	$\begin{bmatrix} d_{11} & . & . & . & . & d_{1n} \\ d_{21} & . & . & . & . & d_{2n} \\ d_{31} & . & . & . & . & d_{3n} \end{bmatrix}$
FOD	VVN	VN	N	F	VF	VVF	.	.	.	.	.	
ROD	VVN	VN	N	F	VF	VVF	$p(d_{31})$	.	.		$p(d_{3n})$	

Similarly the robot's turning fuzzy probability rule is presenting below:

Table 5.3: Probability-fuzzy logic rule for speed clasification

Obstacle's Position	Speed Classification						Probability of Speed Classification						Combination of the Probability of Speed Classification
	S_1	.	.	.	.	S_n	$p(S_1)$	.	.	.	.	$p(S_n)$	C_i
LOD	VVF	VF	F	S	VS	VVS	s_{11}	.	.	.	.	s_{1n}	$\begin{bmatrix} s_{11} & . & . & . & . & s_{1n} \\ s_{21} & . & . & . & . & s_{2n} \\ s_{31} & . & . & . & . & s_{3n} \end{bmatrix}$
FOD	VVF	VF	F	S	VS	VVS	s_{21}	.	.	.	.	s_{2n}	
ROD	VVF	VF	F	S	VS	VVS	s_{31}	.	.	.	.	s_{3n}	

Hence, the turning function of the robot is,

$$T = f\left\{(D, \mu_\theta(D)), (S, \mu_\theta(S)\right\} \tag{5.4}$$

The turning function is a composite function of Distance and Speed, which helps to generate the objective function of a controller. The distance and speed matrix are used to generate a controller. The turning function is represented by the matrix function as follows:

$$T = f\left\{\begin{bmatrix} d_{11} & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & d_{3n} \end{bmatrix}, \begin{bmatrix} s_{11} & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & s_{3n} \end{bmatrix}\right\} \tag{5.5}$$

or,

$$T = f\begin{Bmatrix} ((d_{11},s_{11}),\mu_{11}(d_{11},s_{11})) & . & . & . & . & ((d_{11},s_{11}),\mu_{11}(d_{11},s_{11})) \\ ((d_{21},s_{21}),\mu_{21}(d_{21},s_{21})) & . & . & . & . & ((d_{2n},s_{2n}),\mu_{21}(d_{21},s_{21})) \\ ((d_{31},s_{31}),\mu_{31}(d_{31},s_{31})) & . & . & . & . & ((d_{3n},s_{3n}),\mu_{3n}(d_{3n},s_{3n})) \end{Bmatrix}$$

or,

$$T = f\left\{\begin{bmatrix} A_{11} & . & . & . & . & A_{1n} \\ A_{21} & . & . & . & . & A_{2n} \\ A_{31} & . & . & . & . & A_{3n} \end{bmatrix}\right\},$$

or,

$T = f(p(A)).$

The above set A is studied under the following path optimizing constraint sets:

1. Let, $X = \{p_1, p_2, ..., p_n\}$, where p be the path of the robot. Its subset can be generated as per the probability distribution, if the distribution is discrete, then the subset will be

   $B = \{p_{n-l}, p_{n-m}, ..., p_{n-z}\}$ and for the continuous, the subset will be $C = \{p_j\}$.

2. The membership functions as per the set of obstacles over the set of paths are presented below:

$$\mu_A(o_1) = 1; p_{n-l}$$
$$\mu_A(o_2) = 1; p_{n-m}$$
$$.$$
$$.$$
$$\mu_A(o_m) = 0; p_{n-r}$$
$$.$$
$$.$$
$$\mu_A(o_n) = 1; p_{n-z}$$

3. The optimum path is determined by,

$$\mu_B(o_j) \cap \mu_C(o_j) = \min\{\mu_B(o_j), \mu_C(o_j)\}. \tag{5.7}$$

The execution of algorithm is given below:

1. Let, the set of obstacles be $O = \{o_1, o_2, ..., o_n\}$ and the corresponding set of constraints be $C = \{c_1, c_2, ..., c_m\}$.

2. The decision is $\widetilde{D} = o_1 \cap o_2 \cap ... \cap o_n \cap c_1 \cap c_2 \cap ... \cap c_m$

3. The membership function of the minimized decision is

$$\mu_{\tilde{D}}(X) = \min\left(\mu_{o_1}(o)...\mu_{o_n}(o), \mu_{c_1}(o)...\mu_{c_m}(o)\right) \tag{5.8}$$

4. The membership function of a maximized decision is $x_{\max} = \{o | \mu, \widetilde{D}(X)\}$

The formulation mentioned above is executed as per the follows:

1. Initially, the robot is at the stationary state.
2. Robot traces the goal.
3. Robot achieves the goal when there is obstacle,

Otherwise,

3.1. Robot measures θ.

3.2. Apply $T = f(p(A))$, under the following optimizing function:

$$\mu_D(o) = \mu_G(o) \wedge \mu_C(o) = \min(\mu_G(o), \mu_C(o)), \tag{5.9}$$

$$or, D = G_1 \cap ... \cap G_n \cap C_1 \cap ... \cap C_n,$$

$$or, \mu_{G_1} \wedge ... \wedge \mu_{G_n} \wedge \mu_{C_1} \wedge ... \wedge \mu_{C_n}.$$

By Minimizing,

$$\mu_D(o^*) = \min(\mu_{G_1}(o) \wedge ... \wedge \mu_{G_n}(o) \wedge \mu_{C_1}(o) \wedge ... \wedge \mu_{C_n}(o)), \tag{5.10}$$

$$\text{Or,} \qquad \mu_{D^*}(o) = \begin{cases} \max \mu_D(o), o \in K \\ 0, elsewhere \end{cases}$$

3.3. Robot can reach to the goal with either

$\mu_{G_1}(o)$,

or $\mu_{G_2}(o)$,

.

.

.

$or \mu_{G_n}(o)$,

.

.

.

Corresponds to the following constraints,

$\mu_{C_1}(o)$,

.

.

.

or $\mu_{C_n}(o)$.

Such that, the following transformation function is satisfied,

$$f : A \to A \tag{5.11}$$

Although, the hypothetical formulation is generated and justified by the functional postulates but it is necessary to explain the experimental performance of the proposed model in the domain of real-time. Hence, the analysis of real-time mechanism of the proposed MRN is described below:

5.5 Real Time Analysis of Navigation Mechanism

5.5.1. Obstacle Avoidance and Target Seeking:

It starts with the analysis of the mechanism of the robot. The robot can move 360^o. It has three wheels, 1 in front and 2 in the back. The direction of turning is proportional to the speed difference of back wheels. There may be randomized obstacles, in left, in right and the front. Robot identifies these obstacles with the help of ultrasonic and infrared sensors. Let, there is a smooth floor consisting vertical type obstacles in the environments. Hence, the objective of navigation can be organized under as follows:

1. Robot starts.
2. Defined goals.
3. Identified the obstacles.
4. Turned and moved the robot.
5. Reached in goal.
6. Robot stops.

Apart from the above said steps, robot must follow the necessary action as given below:

1. The path should be free from any collision.
2. The robot should not be hit with any obstacle.
3. The robot should not be hit with any wall.

The performance of the robot depends on the following system

1. Input.
2. Crisp Set.
3. Fuzzifier.
4. Probabilistificaion.
5. Probability-fuzzy Logic controller.
6. Defuzzifier.
7. Deprobabilistificaion.
8. Crisp set.
9. Output.

The membership function set contains six elements of the membership function for each. More precisely it can be constructed as:

1. Triangular membership function.
2. Trapezoidal-triangular membership function
3. Gaussian membership function.

And its elements are the presenting below as the membership function corresponding to each.

1. Very Very Near (VVN)
2. Very Near (VN)
3. Near (N)
4. Far (F)
5. Very Far (VF)
6. Very Very Far (VVF)

The membership function i.e triangular, Trapezodial triangular and Gaussian belongs to the geometrical characteristics and its alements i.e Very Very Near, Very Near, Near Far, Very Far, Very Very Far belongs to linguistic characteristics. These are not sufficient decision variables. For more such variables, it requires obtaining information from the sensors of the robots. Hence, these variables are formed as a function of the probability distribution and the membership function for executing navigation. The probability applied in fuzzy for the distribution of range and both discrete and continuous distribution is used along with the fuzzy parameter. Basically, at the initial level, the discrete distribution is used but for the middle and final level performances both distributions are applied. In the environment, there is uncertainty about the path, obstacle, dimension, decision, the function of velocity measurement but the presentation of this uncertainty is transformed into the constrained by the probability.

In an unknown cluttered environment, a real-time navigation based on probability-fuzzy logic is proposed in this chapter. The objective of the robotics is unique, which is to reach the goal by the optimized trajectory without collision with the obstacles. The LV and RV are the Left Wheel Velocity and Right Wheel Velocity respectively used for to control the direction of trajectory. LOD for Left Obstacle Distance, ROD for Right Obstacle Distance and FOD for Front Obstacle Distance are used as the key term in this chapter. The VVN and VVF are eliminated due to efficiency reason.

The fuzzy parameter and Probability-fuzzy parameter over distance and heading angle is shown in the following table as:

Table 5.4: Fuzzy logic parameters for obstacles:

Linguistic Variable	Very Near (VN)	Near (N)	Medium (M)	Far (F)	Very Far (VF)
LOD	0.0	0.2	0.4	0.6	0.8
ROD	0.2	0.4	0.6	0.8	1.0
FOD	0.4	0.6	0.8	1.0	1.2

Table 5.5: Probability-fuzzy logic parameters for ostacles:

Linguistic Variable	P (VN)	P (N)	P (M)	P (F)	P (VF)
LOD	0.0/0.6	0.2/1.2	0.4/1.8	0.6/2.4	0.8/3.0
ROD	0.2/0.6	0.4/1.2	0.6/1.8	0.8/2.4	1.0/3.0
FOD	0.4/0.6	0.6/1.2	0.8/1.8	1.0/2.4	1.2/3.0

Table 5.6: Fuzzy logic parameters for heading angle:

Linguistic Variable	MN	N	Z	P	MP
Target Heading Angle	-180	-120	-10	10	60
	-120	-60	0.0	60	120
	-60	0	10	120	180

Table 5.7: Probability-fuzzy logic parameters for heading angle:

Linguistic Variable	P (MN)	P (N)	P (Z)	P (P)	P (MP)
Target Heading Angle	-180/360^0	-120/180^0	-10/20^0	10/190^0	60/360^0
	-120/360^0	-60/180^0	0.0/20^0	60/190^0	120/360^0
	-60/360^0	0/180^0	10/20^0	120/190^0	180/360^0

The left and right wheel velocities have the five membership function as Very small (VS), Small (S), Medium (M), Fast (F), Very Fast (VF) and the relative angle with the target is considered as More Positive (MP), Positive (P), Zero (Z), Negetive (N) and More Negetive (MN).

There may be existed a condition of “No Target Consider (NTC)”.

If-Then Rule is tabulated in the below for controlling the MRN as follows:

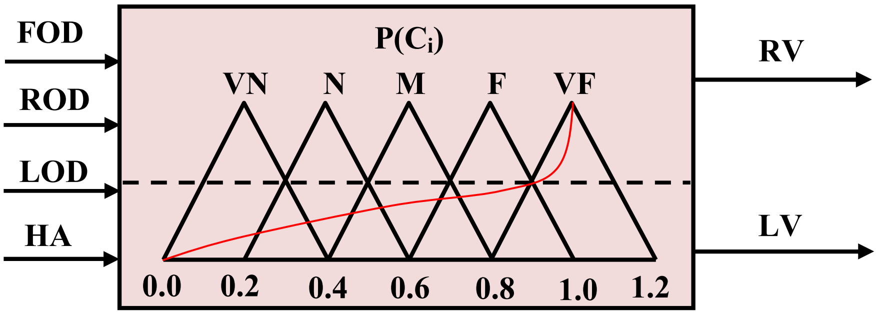


Figure 5.3: Probability-fuzzy logic triangular membership function

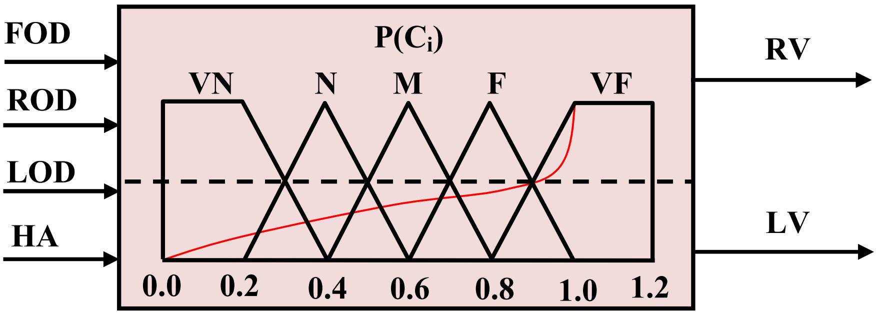


Figure 5.4: Probability-fuzzy logic triangular-trapezoidal membership function

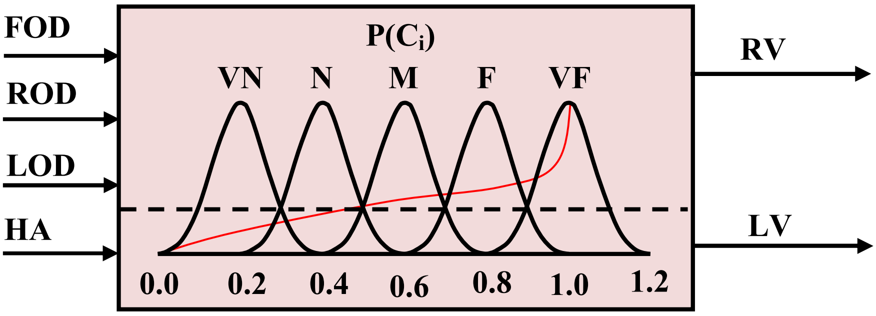


Figure 5.5: Probability-fuzzy logic Gaussian membership function

5.5.2. The Probability-Fuzzy Logic Mechanism for Navigation Control:

The rule comprises with IF-THEN is presented as follows:

Table 5.8 If-Then rule

IF	THEN
$LOD = \frac{LOD_i}{FOD}$	
$FOD = \frac{FOD_i}{ROD}$	$\frac{LV_{ijkl}}{RV_{ijkl}}$
$ROD = \frac{LOD_i}{HA}$	
$HA = \frac{HA_i}{LOD}$	

Where, $i=\frac{1}{5}, j=\frac{1}{5}, k=\frac{1}{5}$ and $l=\frac{1}{5}$

Table 5.9 The Probability-fuzzy If-Then rule is generalized as below:

IF	THEN
$\frac{LOD}{LOD_i} \wedge \frac{FOD}{FOD_j} \wedge \frac{ROD}{ROD_k} \wedge \frac{HD}{HD_l}$	$\frac{LV}{LV_{ijkl}}$
$\frac{LOD_i}{LOD} \wedge \frac{FOD_j}{FOD} \wedge \frac{ROD_k}{ROD} \wedge \frac{HD_l}{HD}$	$\frac{RV}{RV_{ijkl}}$

The compact rule is presented by:

$$W_{ijkl} = dis_i\left(\frac{\mu_{LD_i}}{L_{\mu_{LD_i}}}\right) \wedge\ dis_j\left(\frac{\mu_{RD_j}}{L_{\mu_{RD_j}}}\right) \wedge\ dis_k\left(\frac{\mu_{HA_k}}{L_{\mu_{HA_k}}}\right) \wedge\ (ang_l) \tag{5.12}$$

Where, the measured distance are $dis_i, dis_j, dis_k, dis_l$ and the heading direction is (ang_l).

Let, the vel_{LV} be the velocity of the left wheel and vel_{RV} be the velocity of right wheel, then the following are the rules for MRN.

$$(vel)^{\mu}\, LV'_{ijkl} = \frac{W_{ijkl}}{(vel_{LV})\ \mu_{LV_{ijkl}}}\ ;\ vel\ \epsilon\ LV \tag{5.13}$$

And

$$(vel)^{\mu}\, RV'_{ijkl} = \frac{W_{ijkl}}{(vel_{RV})\ \mu_{RV_{ijkl}}}\ ;\ vel\ \epsilon\ RV \tag{5.14}$$

Hence, the final rule for all membership function is combined as follows:

$$(vel)^{\mu} = \frac{W_{ijkl}}{(vel_{RV})\ \mu_{RV_{ijkl}}}\ ;\ vel\ \epsilon\ RV \tag{5.15}$$

$$(vel)\mu_{LV} = \frac{(vel_{LV})\mu_{LV'1111}}{(vel_{LV})\mu_{LV'ijkl}} \vee \frac{(vel_{LV})\mu_{LV'5555}}{(vel_{LV})\mu_{LV'ijkl}} \tag{5.16}$$

$$(vel)\mu_{RV} = \frac{(vel_{RV})\mu_{RV'1111}}{(vel_{RV})\mu_{RV'ijkl}} \vee \frac{(vel_{RV})\mu_{RV'5555}}{(vel_{RV})\mu_{RV'ijkl}} \tag{5.17}$$

Finally, the crisp transformation can be presented by the below rule as:

$$LV = \frac{\sum (vel)\frac{\mu_{LV}}{(vel)\mu_{LV}}}{\sum (vel)\mu_{LV}} \tag{5.18}$$

$$RV = \frac{\sum (vel) \frac{\mu_{RV}}{(vel)\mu_{RV}}}{\sum (vel)\mu_{RV}} \tag{5.19}$$

The equation 5.18 and 5.19 is for the discrete probability distribution. For the continuous probability distribution, LV and RV are formulated as below:

$$LV = \frac{\int (vel) \frac{\mu_{LV}}{(vel)\mu_{LV}} d(vel)}{\int (vel)\mu_{LV} \; d(vel)} \tag{5.20}$$

$$RV = \frac{\int (vel) \frac{\mu_{RV}}{(vel)\mu_{RV}} d(vel)}{\int (vel)\mu_{RV} \; d(vel)} \tag{5.21}$$

5.6 Obstacle Avoidance

The relation between the robot and goal is studied in this section over the various obstacles, which is in the random form in the environment. The matrix space is the resultant to analyze of robot and target. The same observation is found for robot and obstacles. The force has existed in both the mentioned conditions. Its reaction on the robot is specified as,

- Robot’s velocity can change.
- Robot’s direction can change.
- Robot’s efficiency can change.

Hence, there are the two main challenges of MRN as discussed above, which is to be analyzed in this section under as followings:

- To study the obstacle avoidance and its control mechanism.
- To develop the system of target seeking and its controller.
- To study the robot’s mechanism and its algorithm for execution.

The graph for Probability-Fuzzy logic MRN controller Mechanism is shown below:

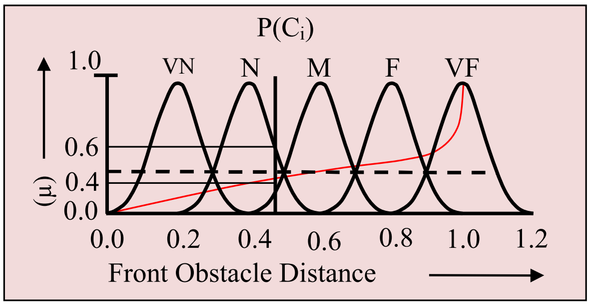


Figure 5.6: Front obstacle distance (FOD)

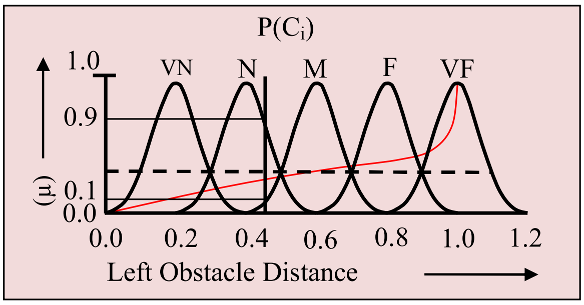


Figure 5.7: Left obstacle distance (LOD)

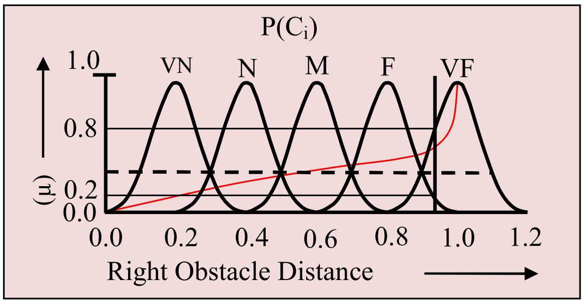


Figure 5.8: Right obstacle distance (ROD)

The rule for Obstacle-Avoidance, target seeking is demonstrated below:

Table 5.10: Probability-fuzzy logic obstacle avoidance (OA) rule

Probability –Fuzzy logic Rule No.	Combination C_i	Action	LD	FD	RD	HA	LV	RV
1	C_1	OA	VN	VN	VN	NTC	VS	S
2	C_2	OA	VF	VN	N	NTC	VS	VF
3	C_3	OA	VN	VN	M	NTC	I	S
4	C_4	OA	VN	VN	F	NTC	F	S
5	C_5	OA	VN	VN	VF	NTC	VF	M
6	C_6	OA	VN	N	VN	NTC	S	S
7	C_7	OA	VN	N	N	NTC	S	S
8	C_8	OA	VN	N	M	NTC	F	M
9	C_9	OA	VN	N	F	NTC	F	S
10	C_{10}	OA	VN	N	VF	NTC	VF	F

Table 5.11: Probability-fuzzy logic target seeking (TS) rule

Probability-Fuzzy logic Rule No	C_i	Action	LD	FD	RD	HA	LV	RV
11	C_{11}	TS	VN	F	N	P/N	S	VS
12	C_{12}	TS	VN	M	VF	P/N	VF	VS
13	C_{13}	TS	N	F	F	P/N	F	S
14	C_{14}	TS	N	F	N	N/P	S	M
15	C_{15}	TS	F	M	N	N/P	M	F
16	C_{16}	TS	F	VF	N	N/P	M	VF

Its steering action is required. Thus, the rule of Probability-fuzzy- logic -steering-action is presented below:

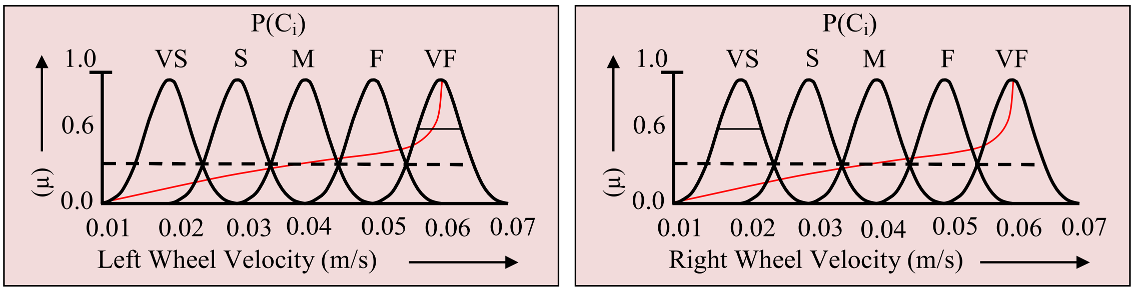


Figure 5.9: Probability-fuzzy logic rule for first combination

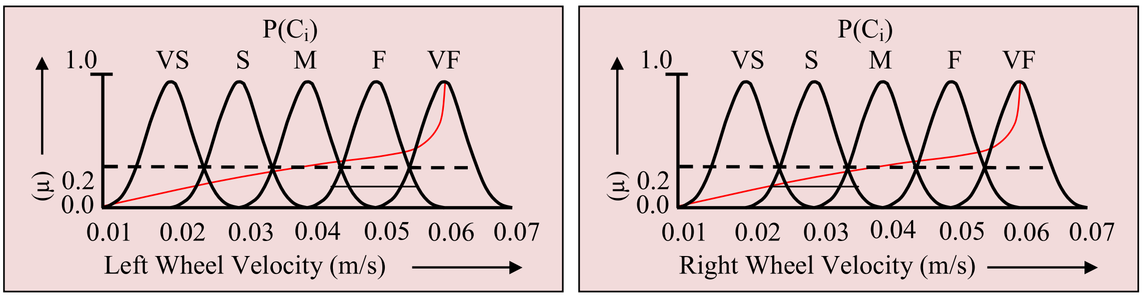


Figure 5.10: Probability-fuzzy logic rule for second combination

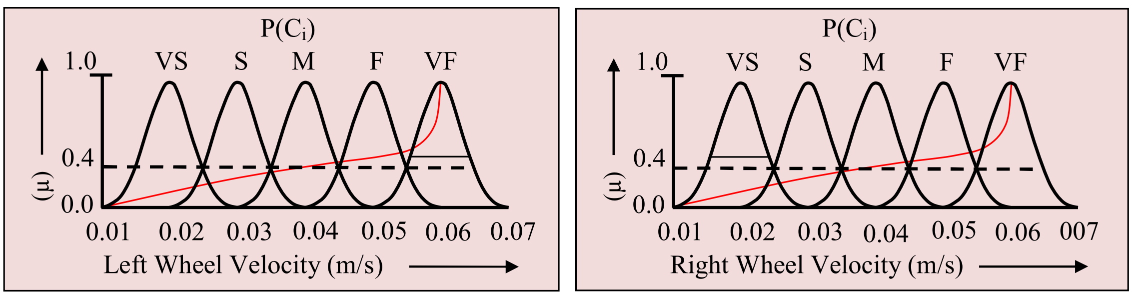


Figure 5.11: Probability-fuzzy logic rule for third combination

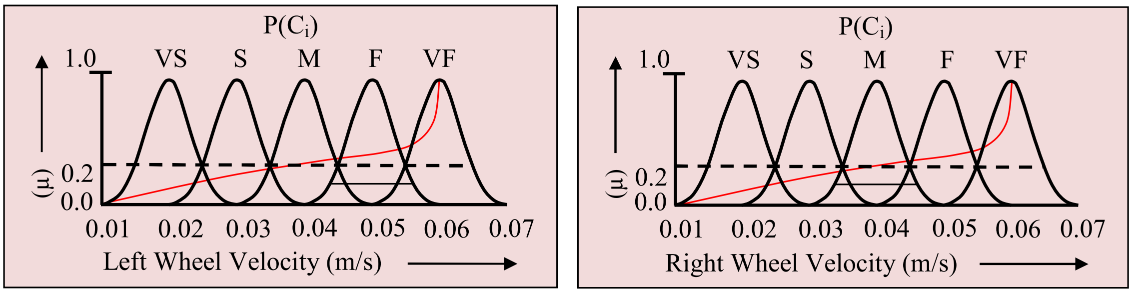


Figure 5.12: Probability-fuzzy logic rule for fourth combination activated

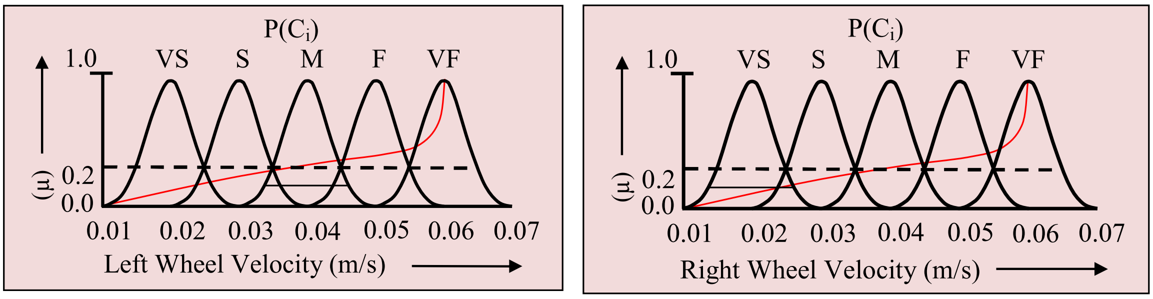


Figure 5.13: Probability-fuzzy logic rule for fifth combination activated

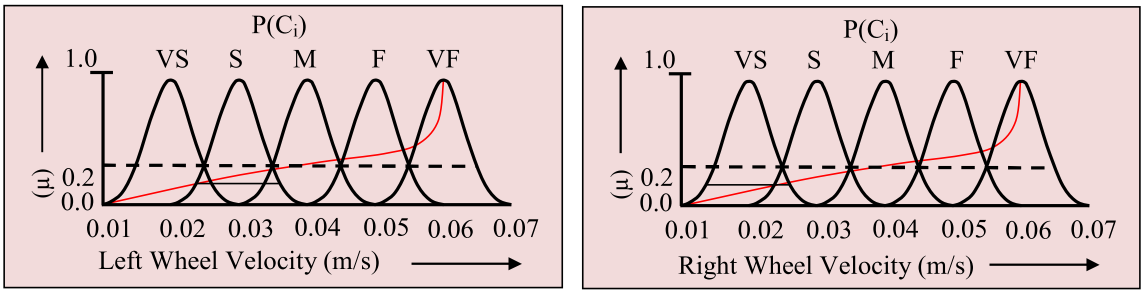


Figure 5.14: Probability-fuzzy logic rule for sixth combination

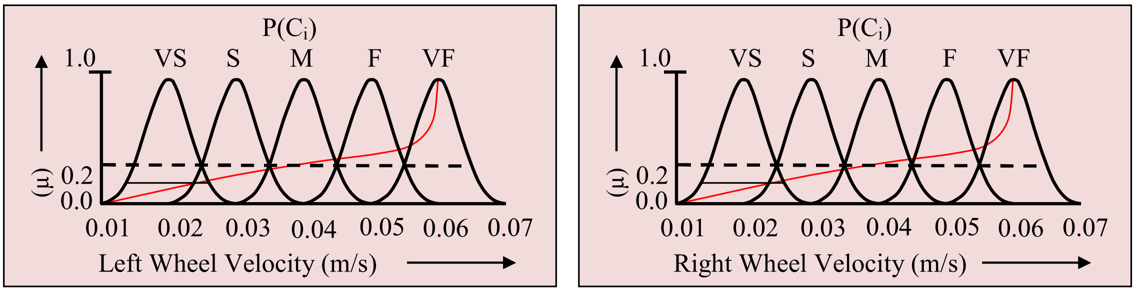


Figure 5.15: Probability-fuzzy logic rule for seventh combination

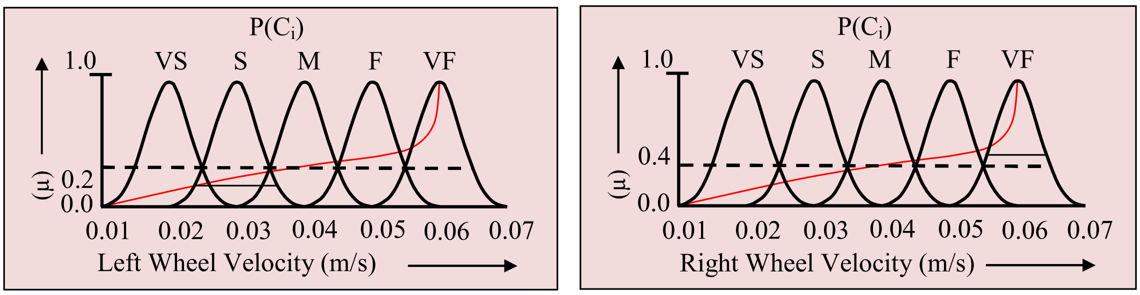


Figure 5.16: Probability-fuzzy logic rule for eighth combination

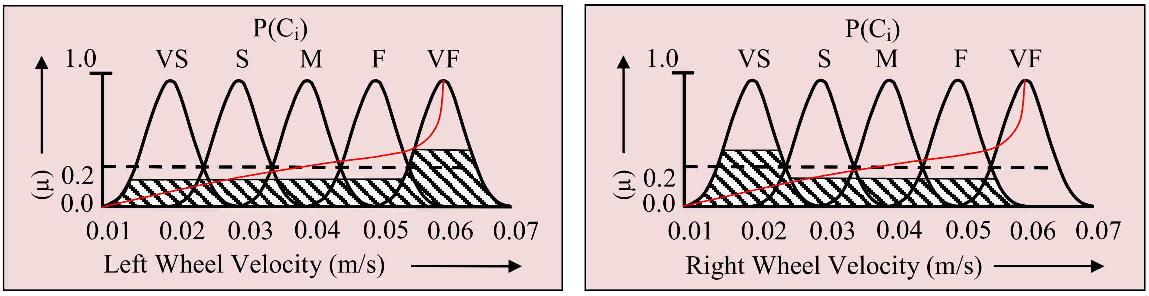


Figure 5.17: Resultant left and right wheel velocity

Table 5.12: Combination table of wheel velocity and obstacle distance

Sl. No.	LOD	FOD	ROD	C_i	LV	RV
1	VN	N	VF	C_1	VF	VS
2	VN	N	F	C_2	F	S
3	VN	M	VF	C_3	VF	VS
4	VN	M	F	C_4	F	M
5	N	N	VF	C_5	M	VS
6	N	N	F	C_6	S	VS
7	N	M	VF	C_7	VF	VS
8	N	M	F	C_8	F	S

5.7 Simulation Analysis

To validate the proposed work, number of tests are conducted in Matlab (R2008) simulation software using probability-fuzzy logic controller. The various environments are created using Matlab software to verify the applicability of the proposed controller for mobile robot navigation. The simulational environment is dealing with the various static and dynamic environments. To check the effectiveness of the proposed controller multiple trials are conducted by changing the position and quantity of obstacles. The single and multiple mobile robots have been considered for the various trials. The different combinations of robot and goal are considered for the simulation analysis such as a single robot with a single goal and multiple robots with a single goal. The simulation results have been tested in 2D space of a 100cm by a 100cm square background in the presence of a variety of static and dynamic obstacles. The Figures 5.18-5.20 demonstrates the efficiency of the PFL navigational controller during obstacle avoidance behavior. The Figure 5.18 consist the start position and one goal position with many obstacles. Initially, robot calculates the navigation angle θ with the goal position and starts accelerating towards the goal without any intelligent mechanism. The Figure shows that when the obstacle comes in the path of the robot, it activities the PFL rules to avoid the obstacle. The process of the obstacle avoidance begins when sensors detect the obstacle and PFL controller activates the PFL rules to control the left and right velocities. The proper heading angle is created by the efficient rules which are given by the probability function. The presented Figure shows the safe path planning of mobile robot in the environment and also shows that the robot creates the safe distance while avoiding the obstacles.

The simulation analysis is also performed in the presence of the dynamic obstacle to present the effectiveness of proposed controller. The Figure 5.21 shows the step by step navigation of mobile robot in the presence of two moving obstacle.

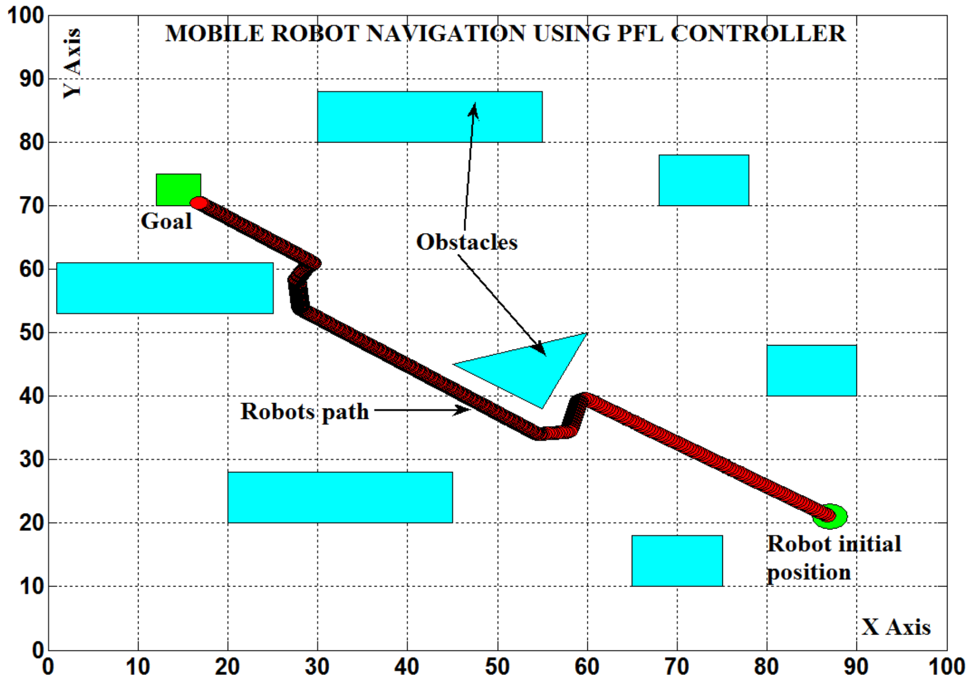


Figure 5.18: Navigation of mobile robot using PFL controller

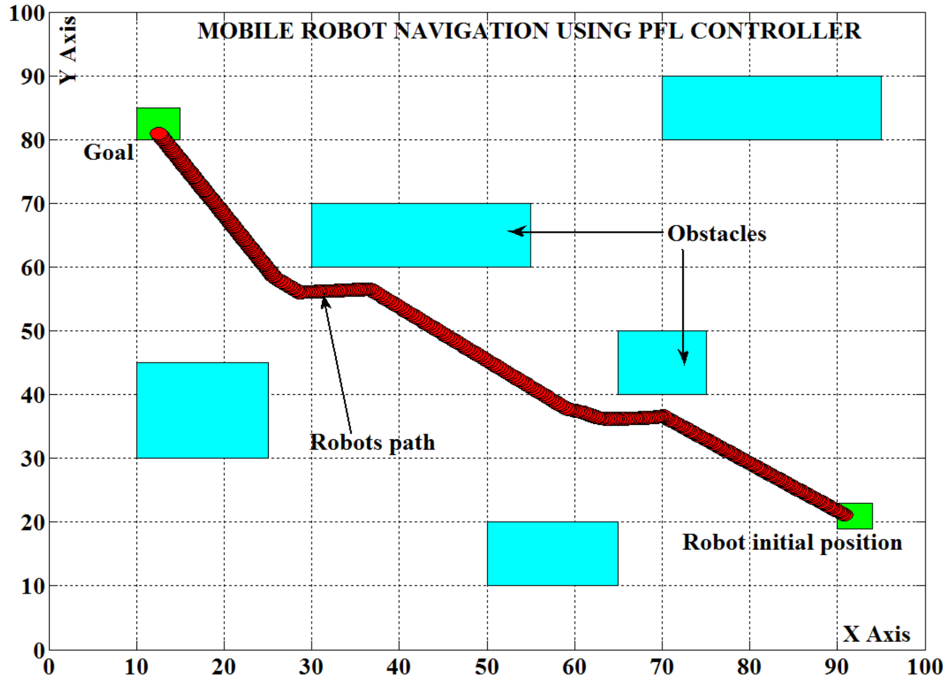


Figure 5.19: Navigation of mobile robot using PFL controller

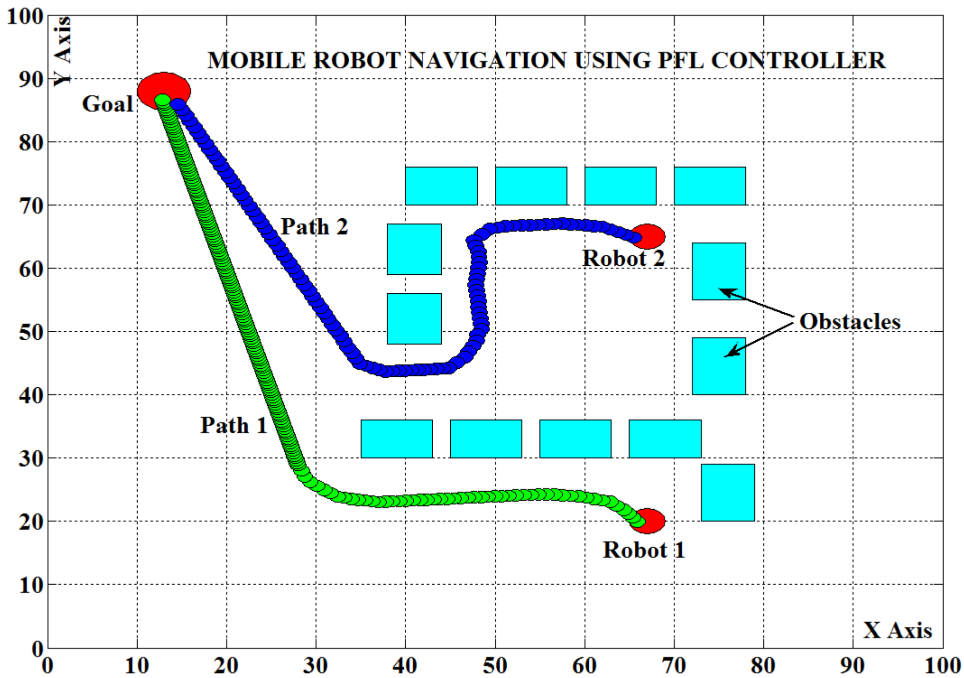


Figure 5.20: Navigation of multiple mobile robots using PFL controller

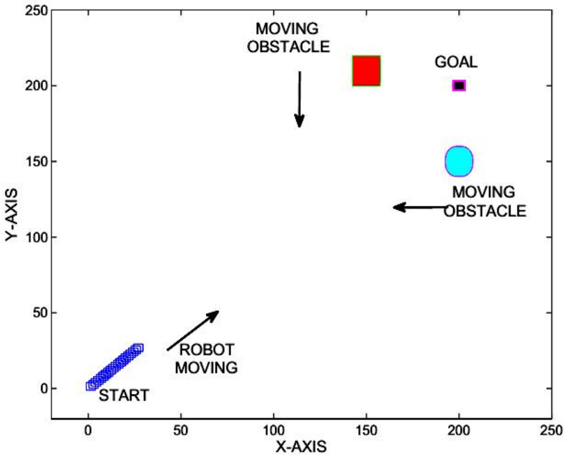


Figure 5.21 (a)

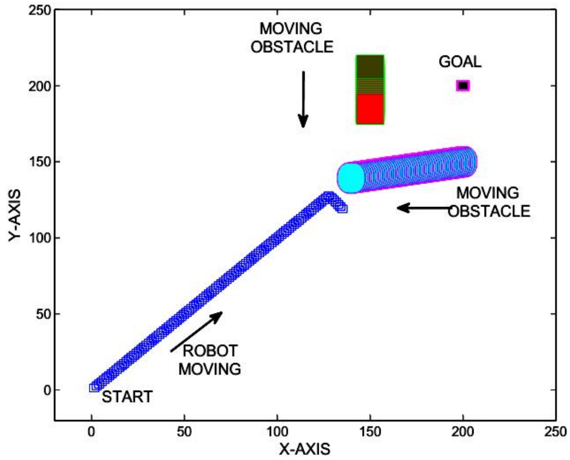


Figure 5.21 (b)

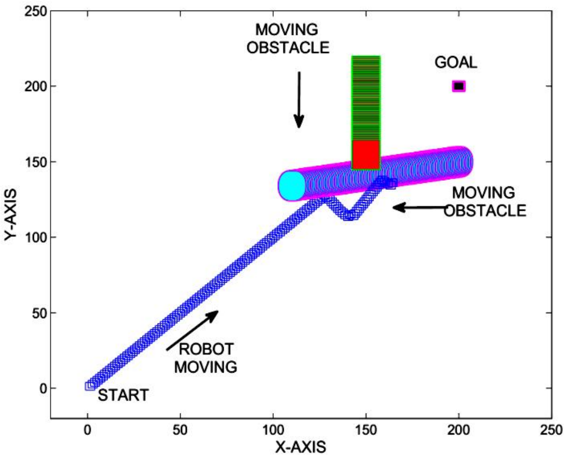


Figure 5.21 (c)

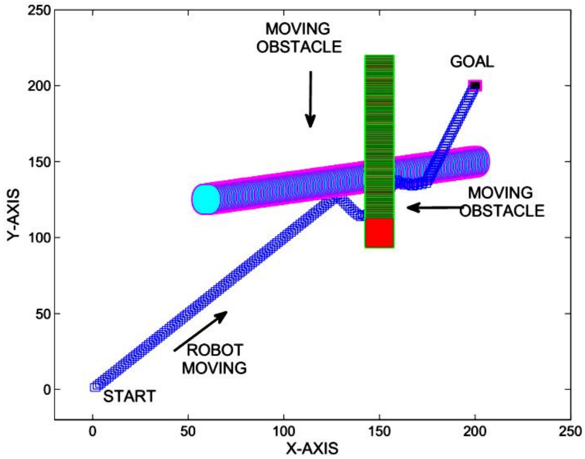


Figure 5.21 (d)

Figure 5.21: Navigation of robot in dynamic environment using PFL controller

5.8 Experimental Analysis

This section proposes the real-time validation of PFL controller for mobile robot navigation. The real time analysis of single and multiple mobile robots in the presence of various obstacles is presented by using the Khepera-II robot. The experiment is conducted on a plane surface. The specification of the robot is mentioned in the Appendix A. The robot uses the infrared sensors to detect the obstacle in short range of 1cm to 5cm. By using the sensory information, robot creates the desired heading angle to achieve the goal.

The Figures 5.22-5.24 show the step by step navigation of the mobile robot from the start point to the goal point. The robot analyzes the environment by using sensory information and then plan towards achieving the goal. The obstacle avoidance mechanism activates when the robot comes in the contact with the obstacle. The PFL based mechanism is uploaded to the robot by using C++ language to get the collision-free path. The proposed controller is very effective as it avoids the random movement in the environment when it detects the obstacles and completes the navigational task in short time. The path smoothness in the below Figure 5.22-5.24 shows the capability of robot to avoid obstacles.

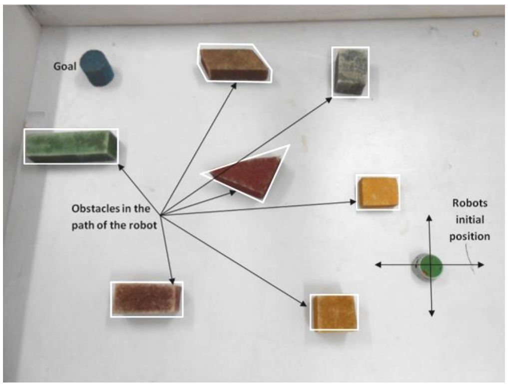


Figure 5.22 (a)

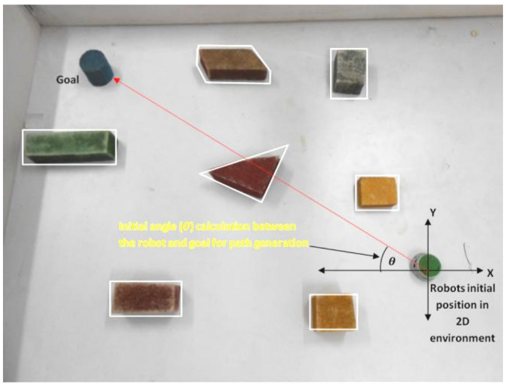


Figure 5.22 (b)

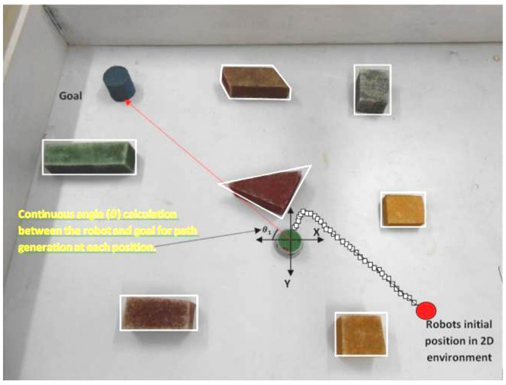


Figure 5.22 (c)

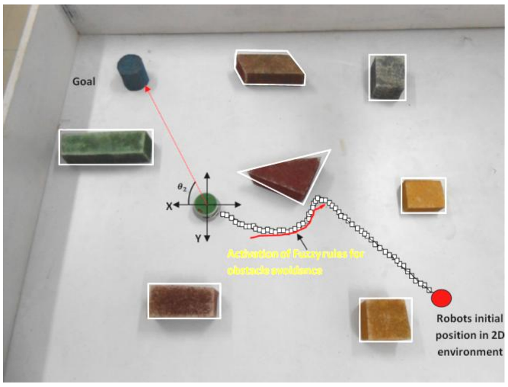


Figure 5.22(d)

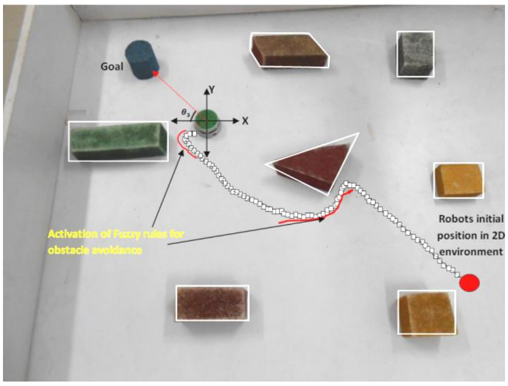


Figure 5.22 (e)

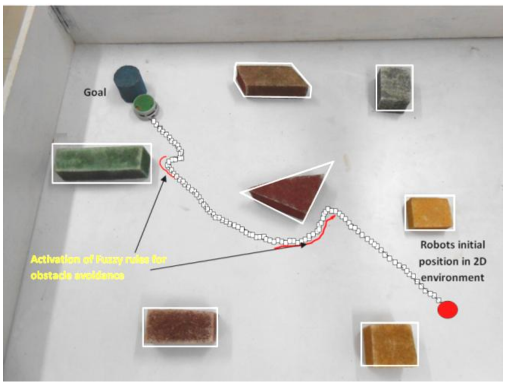


Figure 5.22 (f)

Figure 5.22: Real-time navigation using PFL controller

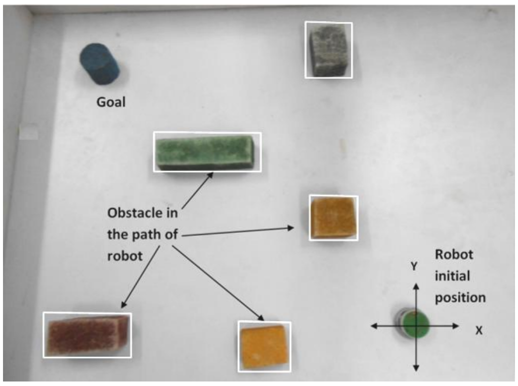


Figure 5.23 (a)

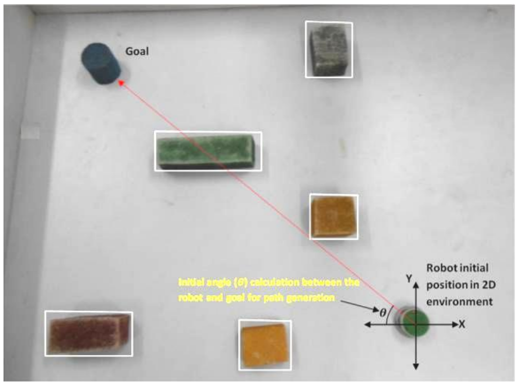


Figure 5. 23 (b)

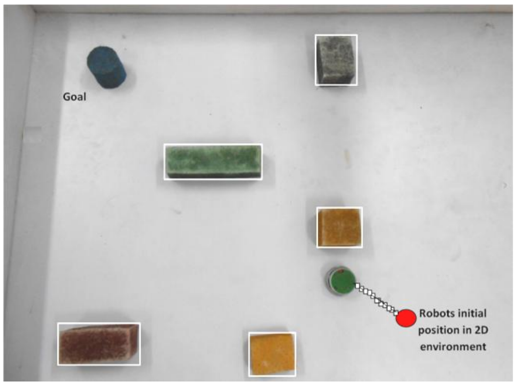


Figure 5. 23 (c)

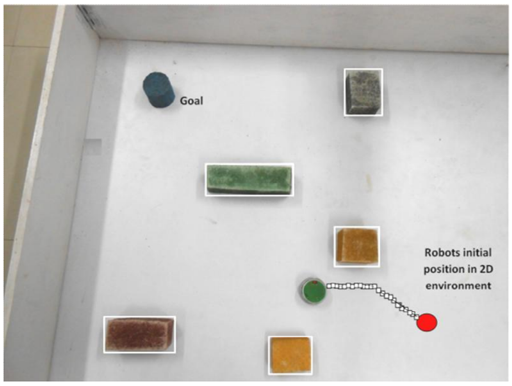


Figure 5. 23 (d)

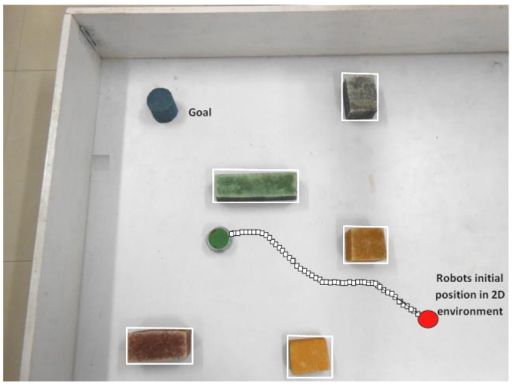


Figure 5. 23 (e)

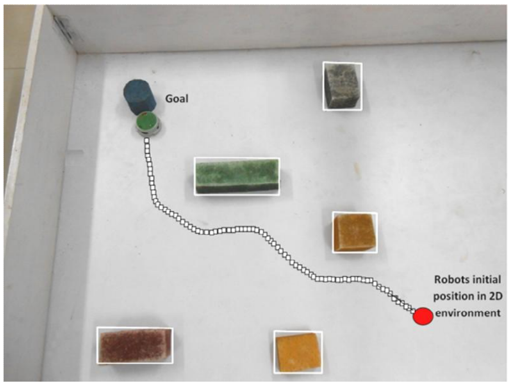


Figure 5. 23 (f)

Figure 5. 23: Real-time navigation using PFL controller

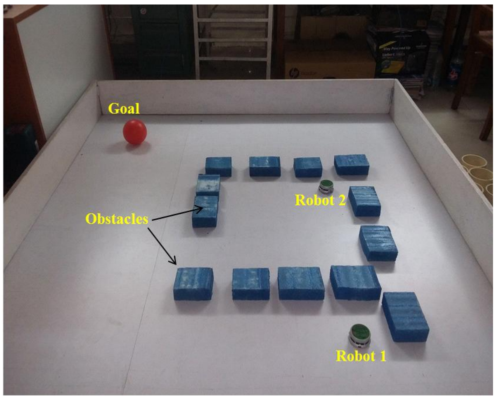


Figure 5. 24(a)

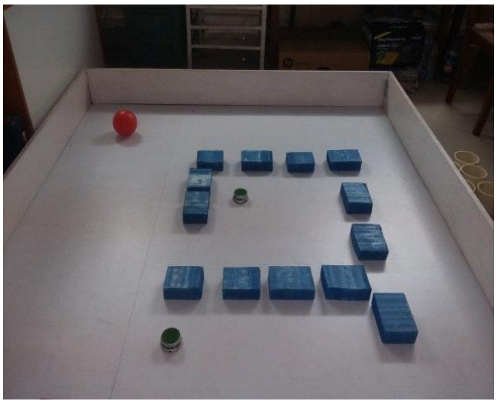


Figure 5. 24(b)

Figure 5. 24(c)

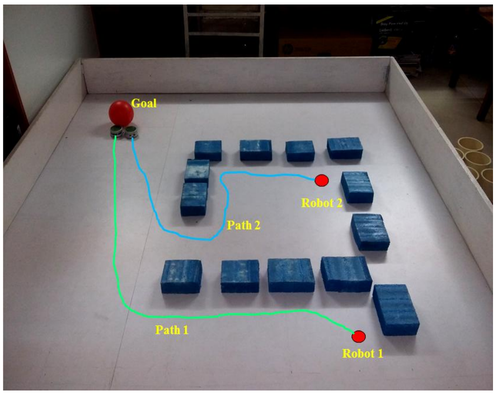


Figure 5. 24(d)

Figure 5.24: Real-time navigation of multiple robots using PFL controller

5.9 Comparative Study of Experimental and Simulation Analysis of MRN over Similar Environment

The comparison between the experimental and simulational analysis is carried over the similar environmental setup to show the workability of the proposed controller. To verify the performance of the robot during the simulation and the real-time experiment, the various trials are performed on the mobile robots. The 20 trials are taken to calculate the path length and required time of navigation for single mobile robot. In Scneraio-1, the Figures 5.18 and 5.22 are compared and in Scenario-2, Figures 5.19 and 5.23 are compared to understand the performance of single mobile robot system. The multiple mobile robot system is compared in Figure 5.20 and 5.24 for navigational path length and required time. The observed path length and required time of MRN for 20 trials are tabulated in Tables 5.11-5.14 for single mobile robot system whereas the 10 trials are tabulated in Tables 5.15-5.16 for multiple mobile robot system. From the tabulation, it is observed that the simulational result is better when compared to experimental results. The observed path length is minimum in case of simulational analysis as compared to experimental analysis for same environmental setup. The obtained percentage of deviation for path length is upto 5.8%. The time required to accomplish the task of navigation in the experimental analysis is more than the simulational analysis. The observed percentage of deviation is less than 6.4%,

Table 5.13: Path length in same simulational and experimental setup (Figure 5.18 and 5.22).

No. of runs	Experimental path length during MRN (in'cm')	Simulational path length during MRN (in 'cm')	% of deviation
1	152.52	144.67	5.14
2	150.12	143.52	4.39
3	150.55	144	4.35
4	154.39	146.37	5.19
5	151.85	142.42	6.21
6	152.64	141.9	7.03
7	150.17	145.64	3.01
8	153.75	143	6.99
9	152.29	146.3	3.93
10	153.9	146.51	4.80
11	154	144.84	5.94
12	153.45	144.25	5.99
13	151.68	142.87	5.80
14	149.59	141.67	5.29
15	154.38	145.2	5.94
16	154.46	145.5	5.80
17	151.98	146.74	3.44
18	153.57	143.17	6.77
19	152.92	141.83	7.25
20	152.82	144.6	5.37
Average path length covered	**152.55**	**144.25**	**5.43**

Table 5.14: Path length in same simulational and experimental setup (Figure 5.19 and 5.23).

No. of runs	Experimental path length during MRN (in'cm')	Simulational path length during MRN (in 'cm')	% of deviation
1	158	150.3	4.87
2	154.34	149.21	3.32
3	156.14	148.61	4.82
4	154.87	149	3.79
5	160.94	148.12	7.96
6	161.48	147.6	8.59
7	158.5	148.94	6.03
8	159.68	150.68	5.63
9	161.39	149.28	7.50
10	155.42	147.9	4.83
11	157.28	147.4	6.28
12	162.2	150.9	6.96
13	158.9	151.24	4.82
14	160.65	152.79	4.89
15	161.15	153.5	4.74
16	162.5	151.6	6.70
17	159.75	151.7	5.03
18	158.59	150.13	5.33
19	155.46	147.6	5.05
20	160.14	148	7.58
Average path length covered	**158.86**	**149.72**	**5.74**

Table 5.15: Navigation time in same simulational and experimental setup (Figure 5.18 and 5.22).

No. of runs	Experimental time during MRN (in 'sec')	Simulational time during MRN (in 'sec')	% of deviation
1	19.83	18.93	4.53
2	19.42	18.3	5.76
3	19	18.2	4.21
4	18.5	17.4	5.94
5	19.6	18.69	4.64
6	17.95	16.84	6.18
7	17.8	16.5	7.30
8	18.61	17.65	5.15
9	18.32	17.31	5.51
10	17.98	16.62	7.56
11	19.2	17.9	6.77
12	19.8	18.46	6.76
13	18.74	17.74	5.33
14	17.97	17	5.39
15	17.54	16.61	5.30
16	18	16.7	7.22
17	18.35	17.22	6.15
18	18.85	17.41	7.63
19	17.58	16.2	7.84
20	17.97	16.77	6.67
Average time required	**18.55**	**17.42**	**6.09**

Table 5.16: Navigation time in same simulational and experimental setup (Figure 5.19 and 5.23).

No. of runs	Experimental time during MRN (in 'sec')	Simulation time during MRN (in 'sec')	% of deviation
1	20.3	19	6.40
2	19.12	18.03	5.70
3	19.27	18.13	5.91
4	19.68	18.24	7.31
5	20.52	19.1	6.92
6	20.41	19.38	5.04
7	20.3	19.2	5.41
8	20.78	19.36	6.83
9	19.95	18.57	6.91
10	19.8	18.23	7.92
11	20.8	19.41	6.68
12	20.45	18.98	7.18
13	19.7	18.17	7.76
14	19.76	18.55	6.12
15	19.75	18.91	4.25
16	19.84	18.88	4.83
17	20.14	19.13	5.01
18	20.77	19.67	5.29
19	19.98	18.42	7.80
20	21.25	19.7	7.29
Average time required	**20.11**	**18.85**	**6.33**

Table 5.17: Path length in same simulational and experimental setup (Figure 5.20 and 5.24).

No. of runs	Robot No.	Experimental path length during MRN (in'cm')	Simulational path length during MRN (in 'cm')	% of Deviation
1	Robot 1	171.1	162	5.31
	Robot 2	129.6	122.8	5.24
2	Robot 1	172.22	165.26	4.04
	Robot 2	129.17	121.31	6.08
3	Robot 1	172.2	166.32	3.41
	Robot 2	128.82	121.74	5.49
4	Robot 1	173.36	163.97	5.41
	Robot 2	131	122.94	6.15
5	Robot 1	172.95	165.71	4.18
	Robot 2	130.74	123.84	5.27
6	Robot 1	174.21	167.95	3.59
	Robot 2	128.88	122.65	4.83
7	Robot 1	175.65	163.6	6.86
	Robot 2	132	125.75	4.73
8	Robot 1	175	166.11	5.08
	Robot 2	131.76	124.88	5.22
9	Robot 1	173.87	163.71	5.84
	Robot 2	128.66	124.22	3.45
10	Robot 1	174	163.64	5.95
	Robot 2	129	124	3.87
Average path length covered	**Robot 1**	**173.45**	**164.82**	**4.97**
	Robot 2	**129.96**	**123.41**	**5.03**

Table 5.18: Navigational time in same simulational and experimental setup (Figure 5.20 and 5.24).

No. of runs	Robot No.	Experimental time during MRN (in‘sec’)	Simulational time during MRN (in ‘sec’)	% of Deviation
1	Robot 1	27.15	25.99	4.27
	Robot 2	22	21.02	4.43
2	Robot 1	28.08	26.77	4.67
	Robot 2	21.2784	19.98	6.06
3	Robot 1	28.02	26.54	5.30
	Robot 2	20.89	19.62	6.06
4	Robot 1	27.75	25.82	6.98
	Robot 2	20.82	20.13	3.32
5	Robot 1	28.31	26.775	5.43
	Robot 2	21.11	19.77	6.37
6	Robot 1	28.04	26.94	3.95
	Robot 2	21.07	19.96	5.27
7	Robot 1	28.21	26.87	4.75
	Robot 2	21.23	20	5.83
8	Robot 1	27.87	26.05	6.56
	Robot 2	20.76	19.55	5.82
9	Robot 1	27.94	26.55	4.99
	Robot 2	20.77	19.81	4.60
10	Robot 1	27.58	26.41	4.24
	Robot 2	20.73	20	3.56
Average time required	**Robot 1**	**27.89**	**26.47**	**5.11**
	Robot 2	**21.07**	**19.98**	**5.13**

5.10 Performance Analysis of PFL Controller with other Navigational Controller

It is essential to check the potential of the proposed navigational controller with the other AI controllers. To justify, the effectiveness of probability-fuzzy logic controller, it is compared with the ant colony optimization (ACO) algorithm and particle swarm optimization (PSO) algorithm. The same environmental set up is considered for the simulation analysis. The comparative results are provided regarding the path length. The Table 5.17 shows the path length comparison between the other controller and proposed controller. The observed result of the proposed controller, when compared to ACO algorithm and PSO algorithm, are good enough and saves the path length by 15.17% and 25% respectively. So, we can say that the proposed controller is efficient for dealing the MRN problem.

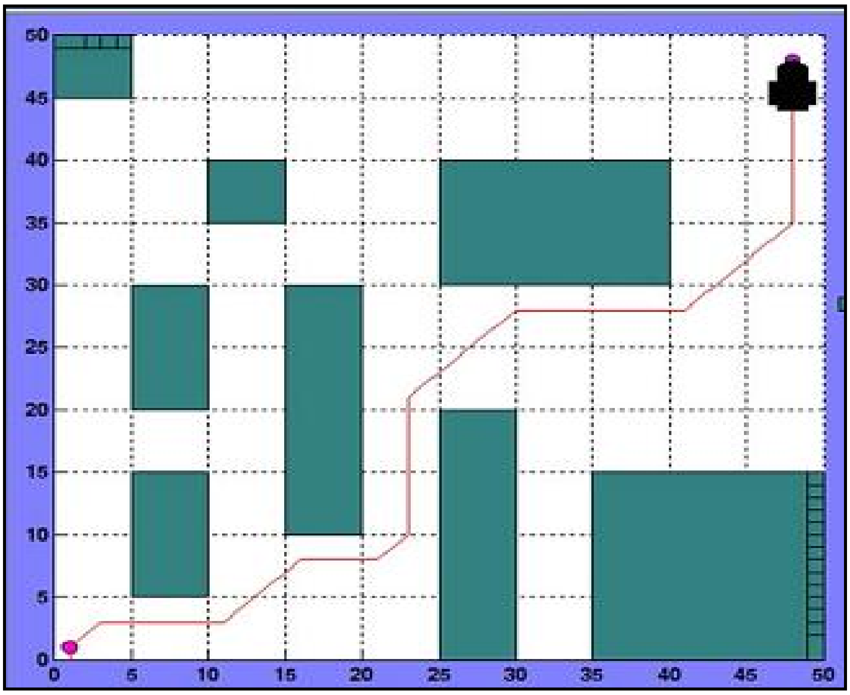


Figure 5.25: Navigation using ACO controller (Garcia et al. [198])

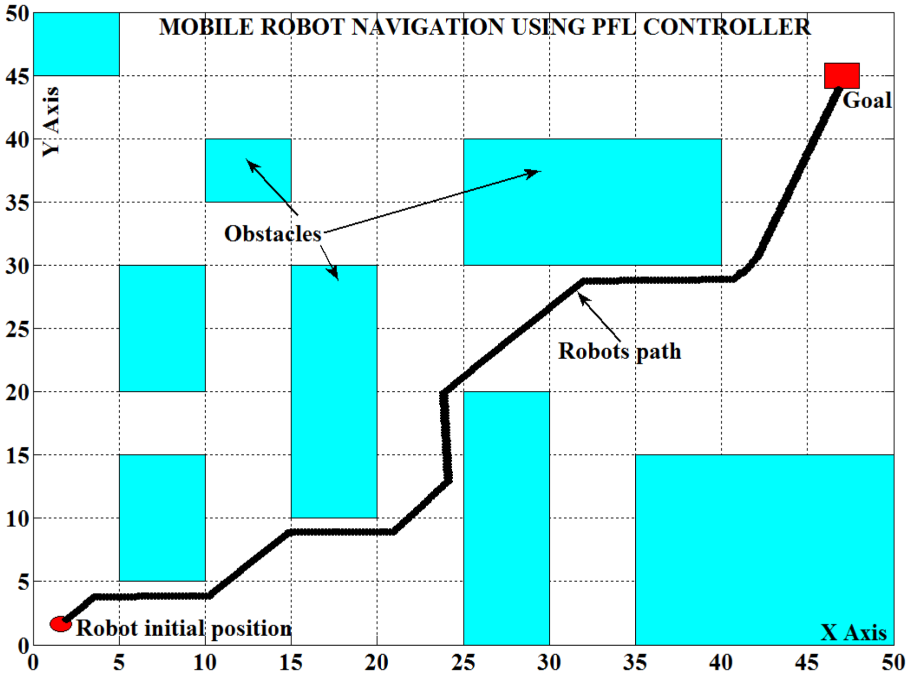


Figure 5.26: Navigation using PFL controller

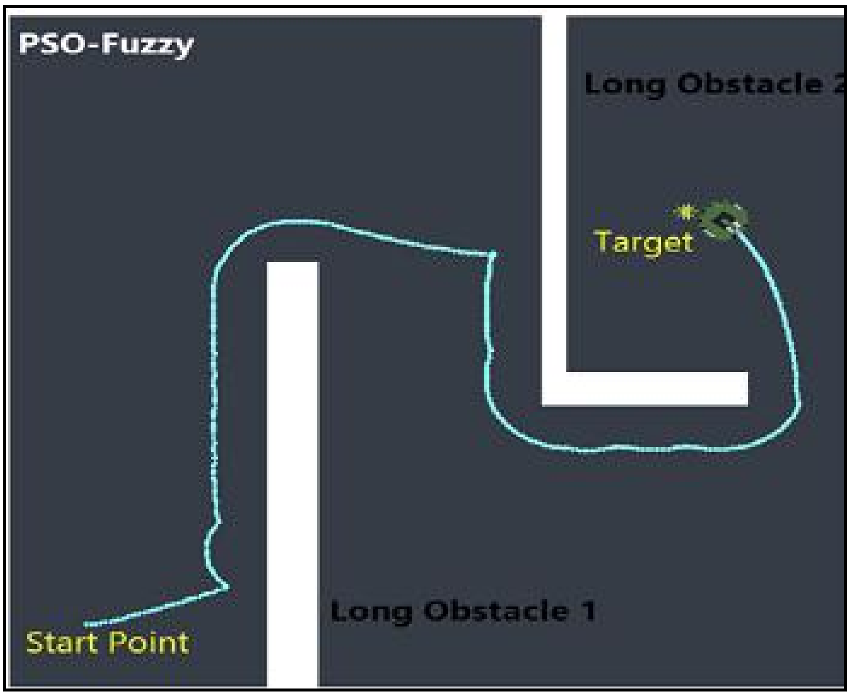


Figure 5.27: Navigation using PSO controller (Algabri et al. [199])

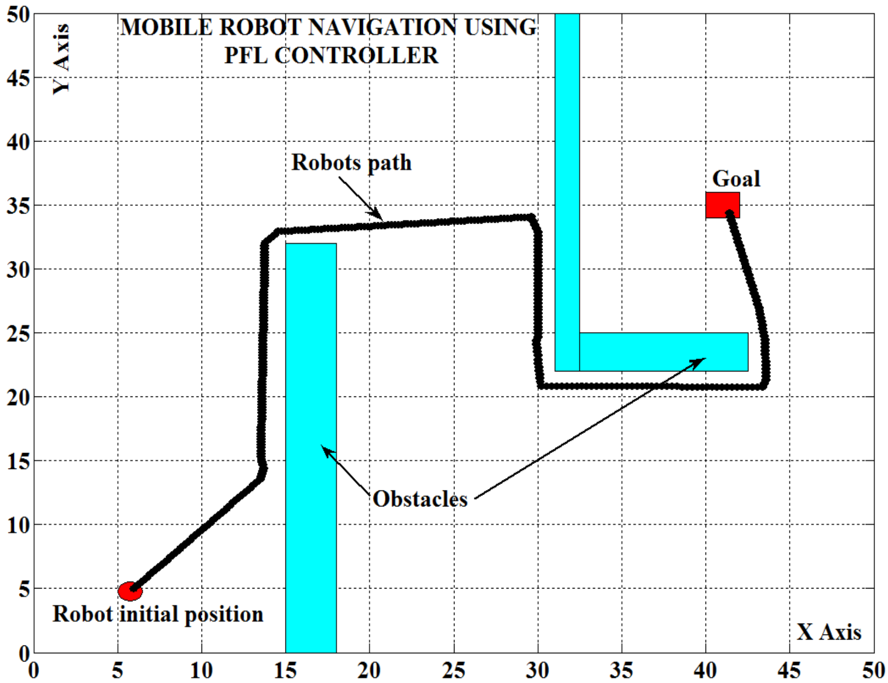


Figure 5.28: Navigation using PFL controller

Table 5.19: Comparision of AI controller with proposed controller regarding path length.

Sl. No.	Simulational path length (in 'cm') using PFL controller	Simulational path length (in 'cm') using other AI controller	% Path length saved using proposed PFL algorithm
Scenario-1	9.5 (Figure 5.26)	11.2 (Figure 5.25)	15.17
Scenario-2	12 (Figure 5.28)	16 (Figure 5.27)	25

5.11 Summary

This chapter presents the application of fuzzy logic via probability distribution for mobile robot navigation. The proposed probability-fuzzy logic decision controller studies the geometric constraints present in the environment. It classifies the decision to avoid obstacles and to generate the efficient path. The valuable findings of the present study from the observed results are:

- The application of Probability with fuzzy logic selects the best decision rule from the possibilities during navigation of robot.
- The proposed controller is robust for navigation in a complex-crowded environment in the presence of a static and dynamic obstacle.

- The Proposed controller can be easily implemented with single and multiple robots at the same time.
- The result shows that the controller is very effective while working in the simulational and experimental environment. The percentage of error is not more than 6.4% between the simulational and experimental results of mobile robot regarding path length and navigational time.
- The Figure 5.21 shows that the proposed controller can handle the problem of mobile robot navigation in a dynamic environment. The navigation in the presence of two moving obstacle is the key finding of the work by using PFL controller.
- The problem of multiple mobile robot navigations using PFL controller is solved efficiently by avoiding collision between the multiple robots and obstacles.
- The obtained results show that the performance of the proposed controller with respect to ACO and PSO is far better regarding the path length. By using the PFL controller the 15.17% path length is saved as compare to ACO controller and 25% path is saved when compared to PSO controller over same environment.

Chapter 6

Analysis of Firefly Algorithm for Mobile Robot Navigation

Recently, Autonomous navigation is one of the most emerging areas of research by using nature inspired metaheuristic algorithm. This chapter presents the application of nature inspired metaheuristic algorithm particularly Firefly Algorithm (FA) for mobile robot navigation in the uncertain environment. The newly introduced FA based approach is used to develop a new navigational strategy for mobile robots in the presence of static and dynamic obstacles.

6.1 Introduction

At present, nature inspired metaheuristic algorithm has a key role in design and development of the mobile robot navigation approach. Nature inspired metaheuristic algorithm is used for mobile robot navigation due to their ability to search space on the global platform to give up the diverse solution and to look for the feasible solution in the local region. Genetic Algorithm, Ant Colony Algorithm, Particle Swarm Optimization, Artificial Bee Colony algorithm, Cuckoo Search Algorithm, Bat Algorithm, imvasive weed optimization, Shuffled Frog Leaping Algorithm etc. have been applied to mobile robot navigation. In the present research work, the FA is implemented for mobile robot navigation in a known and unknown environment in the presence of static and dynamic obstacles. The algorithm is implemented to fulfill the desired goal of navigation such as obstacle avoidance and optimal path planning.

6.2 Overview of Firefly Algorithm

FA is very well known metaheuristic algorithm inspired by the flashing behavior of fireflies developed by Yang in [2008]. It is stochastic in nature as it follows the randomness and works on the principle of trial and error for finding the optimal solution in a realistic amount of time. Firefly is winged beetle of family Lampyridae and commonly called as lightning bugs due to their ability to produce light. Over 2000 species of firefly occurs in nature. It produces light by a process of oxidation of Luciferin in the presence of the enzymes Luciferase, which occurs very quickly. This process of producing light is known as bioluminescence and firefly's uses this light to glow without wasting of heat energy. Fireflies use this light for the purpose of selection of a mate, to communicate a certain message and sometimes it also uses for the scaring off animals who try to eat firefly. Most of the fireflies produce short and rhythmic flashes and the pattern shown in these flashes is unique for most of the times for a particular species. The intensity of light of fireflies and the absorption rate of light by air makes fireflies visible at a limited distance. It can be visible up to few hundred meters at night which are sufficient for all fireflies to communicate with each other. The female of the class responds to an individual pattern of flashing light of the male of the same class. The attraction of both male and female firefly totally depends on the following characteristics in the process of selection of a mate.

- The rhythm of the flashlight.
- The rate of flashing of light.
- The amount of time for which the flash of light is observed.

The flashing light of fireflies is used as the objective function which is to be optimized and used to formulate new optimization algorithm. The attraction of one firefly towards the other is possible when the other is having the higher light intensity and this is the basic concept of working on firefly algorithm. It follows the three basic rules while wondering from each other as;

1. All fireflies are unisex and therefore one firefly will be attracted to other fireflies regardless of their sex.
2. Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move towards the brighter one. Also, as the attractiveness is proportional to the brightness, they both decrease as their distance increases. If there is no brighter firefly than a particular firefly, then both will move randomly.
3. The brightness of a firefly is determined according to the nature of the objective function.

6.3 Structure of Firefly Algorithm

Objective function $f(x),\ x = (x_1, x_2, \ldots\ldots\ldots, x_d)^T$

Generate initial population of fireflies $x_i\ (i = 1,2,3,\ldots\ldots\ldots,n)$

Light intensity l_i *at* x_i *is determined by* $f(x_i)$

Define light absorption coefficient γ

While *(t < Max Generation)*

For *i=1: n all n fireflies*

For *j=1: n all n fireflies (inner loop)*

If *($I_i < I_j$), Move firefly I towards j;*

End if

Vary attractiveness with distance r via exp [-γr]

Evaluate new solution and update light intensity

End for *j*

End for *i*

Rank the fireflies and find the current global best g_*

End while

Postprocessor results and visualization

6.4 Basic Parameters of Firefly Algorithm

Attractiveness and light intensity is a key factor while designing firefly algorithm. We can say that the attractiveness of the firefly is determined by its brightness which is the objective function of the algorithm. Let I is the brightness of a firefly and x is its location then we can say that $I(x) \alpha f(x)$. But, the attractiveness β is depends on the distance r_{ij} between the two firefly i and j. The intensity of the light varies as per the distance variation from source to target (r) and the medium of absorption of light. So, light intensity $I(r)$ is given with respect to intensity of source $I(r)$ by using inverse square law is

$$I(r) = \frac{I_s}{r^2} \tag{6.1}$$

and for fixed light absorption coefficient γ, the light intensity I varies with the distance r is

$$I = I_0 e^{-\gamma r} \tag{6.2}$$

Where I_0 is the original light intensity. In order to avoid singularity at $r = 0$ in the expression I_s / r^2, the combined effect of both the inverse square law and absorption can be approximated as the following Gaussians form

$$I(r) = I_0 e^{-\gamma r^2} \tag{6.3}$$

Attractiveness of firefly is proportional to the light intensity which is defined by β as

$$\beta = \beta_0 e^{-\gamma r^2} \tag{6.4}$$

Where β_0 is the attractiveness at $r = 0$. As it is often faster to calculate $\frac{1}{1+r^2}$ than an exponential function, the above function, if necessary, can conveniently be approximated as

$$\beta = \frac{\beta_0}{1+\gamma r^2} \tag{6.5}$$

The equation (6.4) and (6.5) define characteristics distances $\Gamma = 1/\sqrt{\gamma}$ over which the attractiveness changes significantly from β_0 to $\beta_0 e^{-1}$ for equation or $\beta_0 / 2$ for equation 6.5. In the practical examination, the attractiveness function β_r can be any monotonically decreasing function such as the following form

$$\beta(r) = \beta_0 e^{-\gamma r^m}, \ m \geq 1 \tag{6.6}$$

For the characteristics length becomes

$$\Gamma = \gamma^{-1/m} \to 1, \ m \to \infty \tag{6.7}$$

Conversely, for a given length scale Γ in an optimization problem, the parameter γ can be used as a typical initial value. That is

$$\gamma = \frac{1}{\Gamma^m} \tag{6.8}$$

The distance between the two fireflies i and j at x_i and x_j respectively are the Cartesian distance

$$r_{ij} = x_i - x_j = \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k}\right)^2} \tag{6.9}$$

Where $x_{i,k}$ is the k th component of the spatial coordinate x_i of the i th firefly. In 2-D case, we have

$$r_{ij} = \sqrt{\left(y_i - y_j\right)^2 + (x_i - x_j)^2} \qquad 6.10)$$

The movement of a firefly i is attracted to another more attractive brighter firefly j is determined by

$$x_i = x_i + \beta_0 e^{-\gamma r^2 i,j}\left(x_j - x_i\right) + \alpha\epsilon_i, \qquad (6.11)$$

Where the second term is due to the attraction. The third term is randomization with α being the randomization parameter, and ϵ_i, is a vector of random number drawn from a Gaussians distribution or uniform distribution.

6.5 Objective Function Formulation using FA

In mobile robot navigation problem, flashing behavior of firefly is used to find out the optimal path planning when static and dynamic obstacles surround the robot. The effort has been made here to develop effective path optimization approach for a mobile robot using FA regarding path length and time. Path optimization for navigation is the plan of best parameters to get required output as per the objective which includes challenges such as obstacle detection, obstacle avoidance, to face trap like situation, to avoid the random walk and optimal path generation. While moving in the environment, the information about the surrounding is provided by the sensors which are attached on the robot; it helps to localize the position of the robot in the unknown environment. This sensor provides the information about the shape, size and position of the obstacle and by using this sensory information, robots move towards a goal without collision to the obstacle. To produce the optimal and safe path planning for the mobile robots using FA controller is the goal of the present study. Initially, the navigational path optimization problem is converted into the minimization problem and later on it is expressed as an objective function based on the goal and obstacle position as the desired parameter with the implementation of firefly algorithm. During the process of execution, the localization of the globally brighter firefly in each iteration are chosen and the robot moves to this location in series. During the execution, if there is no obstacle found in the path of a mobile robot then the robot directly finds the goal position without using the artificial intelligent mechanism. The key objectives of the proposed FA based navigational controller are as follows:

i. To design and develop the effective path planning algorithm to avoid obstacle present in the path.
ii. To avoid random moving of the robot in its environment as per the time optimality.
iii. To produce the uniqueness in simulation and experimental result.
iv. To give better performance when compared with the other navigational controller.

Fig. 6.1 shows the complete architecture of the proposed FA based controller for mobile robot navigation.

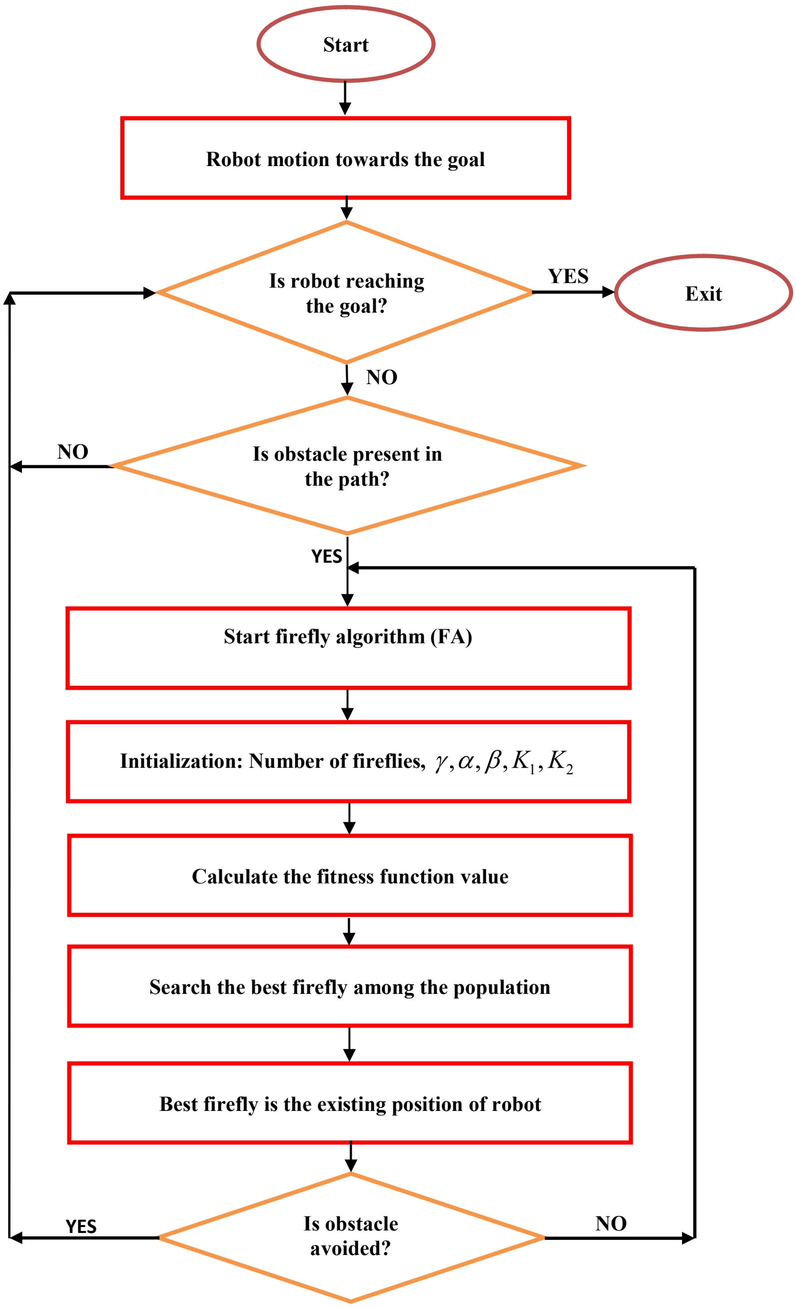


Figure 6.1: Architecture of proposed FA controller for navigation

6.5.1 Obstacle Avoidance Behavior

Navigation is a difficult task for any mobile robot when the environment is uncertain. The uncertainty may be about changing the condition of obstacle or goal. The obstacle present in the environment may be of varying shape and size. To fulfill the task of efficient navigation, the mobile robot requires obstacle avoidance mechanism to avoid collision with the obstacles in the uncertain environment. The firefly algorithm produces the number of random fireflies near the obstacle and the brighter firefly is selected from the group by brightness. The brighter firefly is selected in such a way that, it must be at the extreme safe distance from the nearby obstacle. The robot occupies the position of the newly selected fireflies and the procedure for searching of the next brighter firefly starts till the safe and optimum path generates. The best firefly is selected using Euclidean distance between the best firefly and the closest obstacle which is shown by the equation (6.12) regarding the objective function as below,

Let, D_{fo} stands for the Euclidean distance between the location of firefly with the nearby obstacle, x_{f_i} and y_{f_i} are the x and y coordinates of the firefly position respectively, x_O and y_O are x and y coordinate of obstacle position respectively.

Then, Euclidean distance is

$$\mathrm{D}_{fo} = \sqrt{\left(x_O - x_{f_i}\right)^2 + \left(y_O - y_{f_i}\right)^2} \tag{6.12}$$

For the complex-crowded environment, the selection of the nearby obstacle is must for the optimum path generation and hence the distance between the neighboring obstacles is calculated by the equation (6.13).

Let, D_{RO} is the distance between the robot to nearby obstacle, x_{O_n} and y_{O_n} are the x and y coordinates of the nearby obstacle respectively, x_R and y_R are the x and y coordinates of the robot position respectively.

Then the distance between the robots to nearby obstacle is;

$$\mathrm{D}_{RO} = \sqrt{\left(x_{O_n} - x_R\right)^2 + \left(y_{O_n} - y_R\right)^2} \tag{6.13}$$

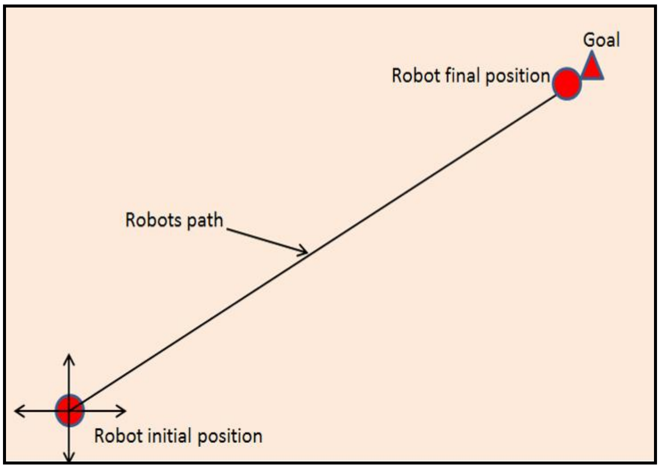


Figure 6.2: Navigation of robot in obstacle free environment

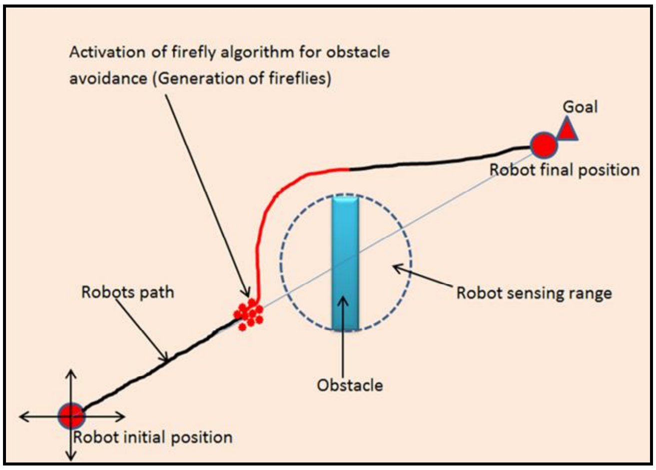


Figure 6.3: Navigation in presence of obstacle using FA controller

6.5.2 Goal Searching Behavior

Here, the brighter firefly is selected from the group of random fireflies in such a way that it has the maximum distance from the obstacle (mentioned in the obstacle avoidance behavior) and a minimum distance of the same firefly from the goal. It is a continuous searching process for brighter firefly over the period till it completely finds the goal. The position of the current brighter firefly must always be at least distance from the goal position. The equation (6.14) shows the Euclidean distance between the goal and firefly.

Let, D_{fg} is the minimum Euclidean distance of firefly with goal position, x_g and y_g are the x and y coordinate of the goal position.

Then, the distance between fireflies with goal is

$$D_{fg} = \sqrt{\left(x_g - x_{f_i}\right)^2 + \left(y_g - y_{f_i}\right)^2} \tag{6.14}$$

From the study, the obstacle avoidance behavior and goal searching behavior comprises to formulation of objective function of fireflies for path planning optimization problem is represented as follows

$$f_i = K_1 . \frac{1}{\min o_n \in o_s \left\| D_{fo} \right\|} + K_2 . \left\| D_{fg} \right\| \tag{6.15}$$

As per the objective function formulation, the environment is considered as '*n*' obstacle and represented as O_1, O_2, O_3, O_4 … O_n and their coordinate position are (x_{o1}, y_{o1}), (x_{o2}, y_{o2}), (x_{o3}, y_{o3}), (x_{o4}, y_{o4}),…… (x_{on}, y_{on}). The number of the obstacle (O_s) is detected by the robot when it comes in threshold range of sensor during navigation in the environment. When the number of fireflies (f_i) lies away from the obstacle, then the value of $\min o_n \in o_s \left\| D_{fo} \right\|$ becomes huge in the objective function and when the f_i lies closer to goal, then the value of $\left\| D_{fg} \right\|$ becomes reduced. Therefore, the study of the objective function by using firefly algorithm comes under the minimization optimization problem which helps to find optimal path planning for mobile robot navigation in an uncertain environment. Here, K_1 is the fitting parameter which decides the path safety; K_2 decides the maximum and minimum path length of the navigation. When the value of K_1 is maximum, then the robot can safely avoid the obstacle without hitting the boundary, however, the chances of collision to obstacle increases with a decrease in value of K_1. The parameters K_1 and K_2 are the constants used in firefly algorithm. Hence, the proper selection of control parameter over the local minima problem decides the success of objective function for robot path planning. Trial and error method has been used for controlling the parameter of the objective function.

6.5.3 Steps involved in the FA for MRN

1. Initialize the robot, goal and obstacle position.
2. Movement of robot towards the goal till it detects the obstacle.
3. If the obstacle exists in the path, then activate FA.
4. Generate the population of fireflies' randomly.
5. Select the brightest firefly among the population to fit equation (6.15).

6. Move robot towards the current brightest firefly position.
7. Repeat the step 2 to 6 till robot avoids the obtacle.
8. Goal achieved.
9. Stop.

6.6 Simulation Analysis

To analyse the performance of present FA based controller regarding path length and time required for navigation, the variety of environments have been tested in Matlab (R2008) simulation software. The simulation experiment performed on the PC with I3 processor (3GHz), 4 GB RAM, 500 GB hard disk, Windows 7 (64 bit) OS, NVDIA (1GB) graphics card.

To develop the efficient FA controller, the proper selection of the parameter is must according to the problem domain. Randomization parameter (α), Light absorption coefficient (γ) and the attractiveness (β) are the controlled parameter which decides the functioning of FA. These parameters are considered for problem analysis between the ranges 0 to 1. To minimize the computational efforts, the number of fireflies and maximum generation considered for MRN problem. The series of experiment have been conducted by varying control parameter (γ,α,β) with step of 0.05, where number of fireflies with step of 5. The final control parameters are finalized after performing optimization for 50 times and it observed at $\gamma = 0.5$, $\alpha = 0.5$ and $\beta = 0.2$. The obtained control parameter by FA is considered as initial position of the robot during obstacle avoidance of navigational problem.

The simulation results have been tested in 2D space of a 100cm by a 100cm square background in the presence of a variety of static and dynamic obstacle. Figure 6.4-6.6 demonstrate the efficiency of the FA based navigational controller while avoiding the obstacle in the environment. The path obtained by the robot is optimal regarding path length and smooth in the sense of making safe distance with the obstacle.

The simulation analysis in presence of dynamic obstacle is also presented to prove the effectiveness of proposed controller. The Figure 6.7 shows the step by step navigation of mobile robot in the presence of two moving obstacle. At the time of navigation when the robot moves towards the goal, the pink color path shows the activation of FA for the avoidance of dynamic obstacle.

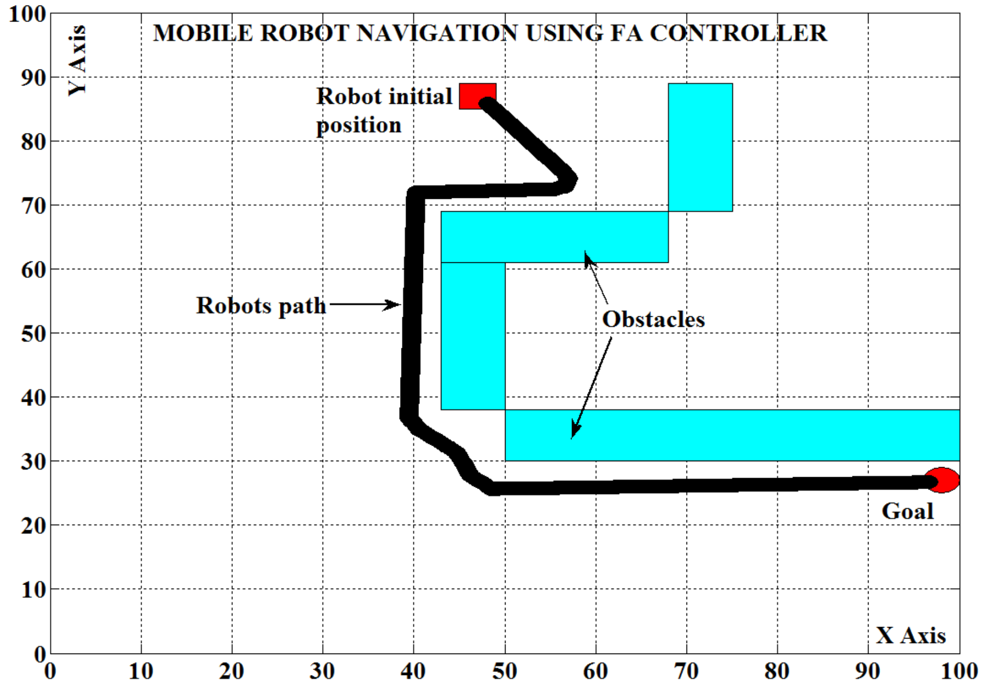


Figure 6.4: Navigation of mobile robot using FA controller

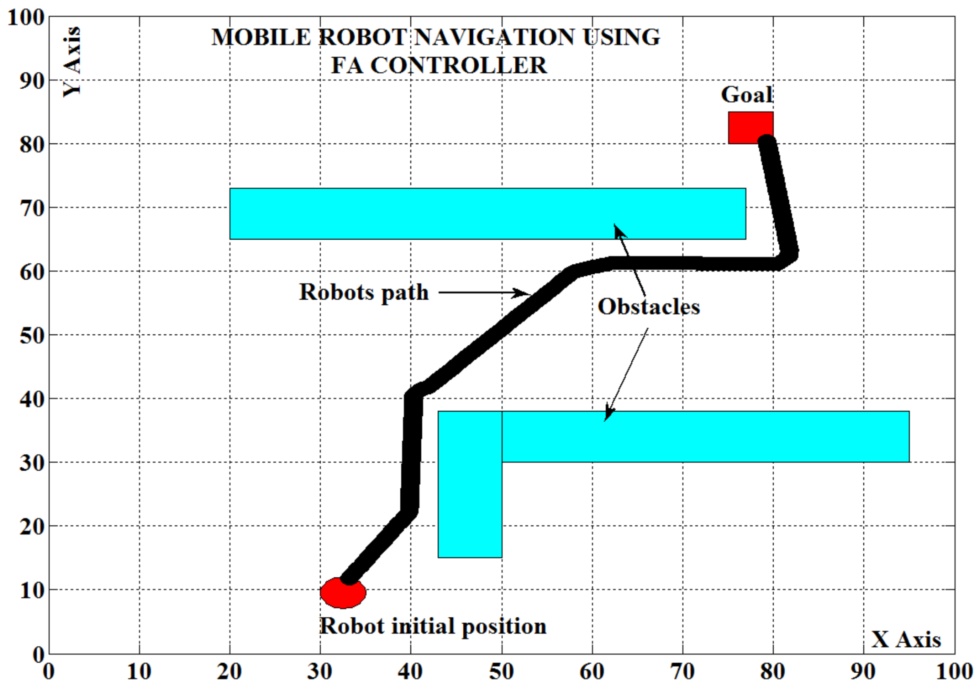


Figure 6.5: Navigation of mobile robot using FA controller

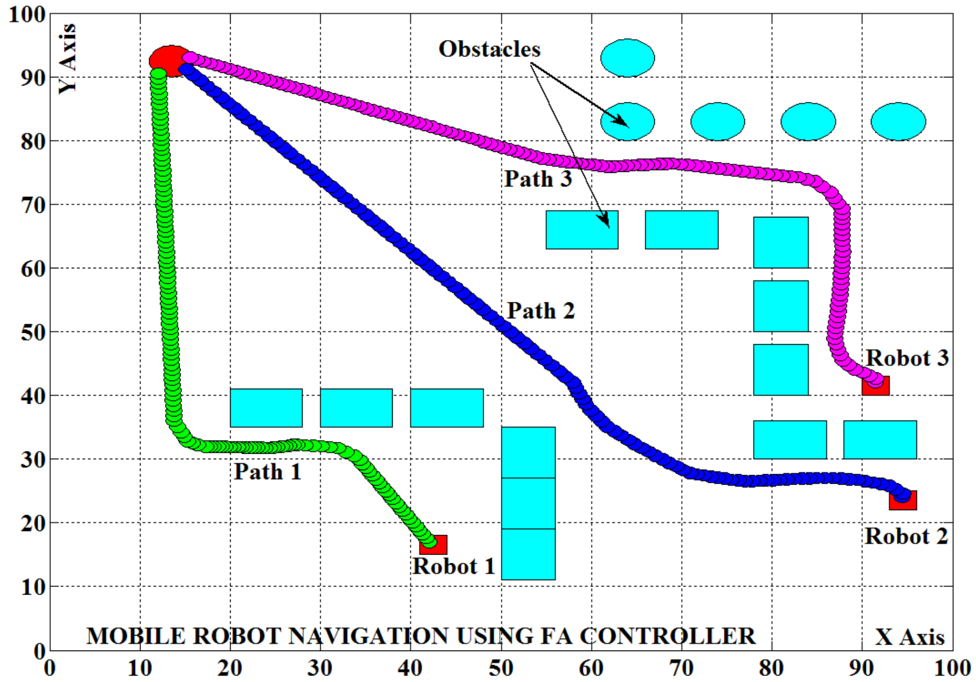


Figure 6.6: Navigation of multiple mobile robot using FA controller

Table 6.1: Parameters for FA

Sl. No.	Parameters	Values in range
1	No. of fireflies (N)	5-100
2	No. of generation	50-100
3	Light absorption coefficient (γ)	0.1-1
4	Randomization parameter (α)	0.1-1
5	Attractiveness (β)	0.1-1
6	Fitting parameter (K_1)	0.1-1
7	Fitting parameter (K_2)	0.01-0.0001

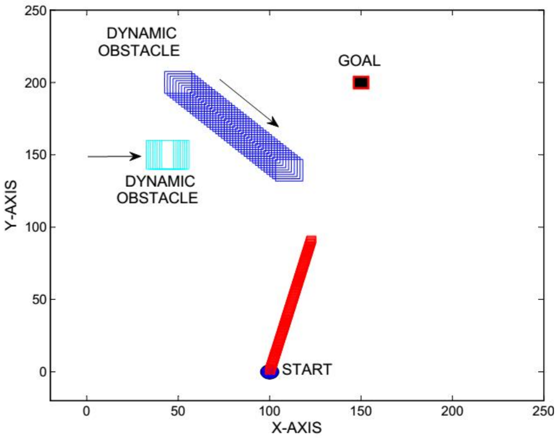


Figure 6.7 (a)

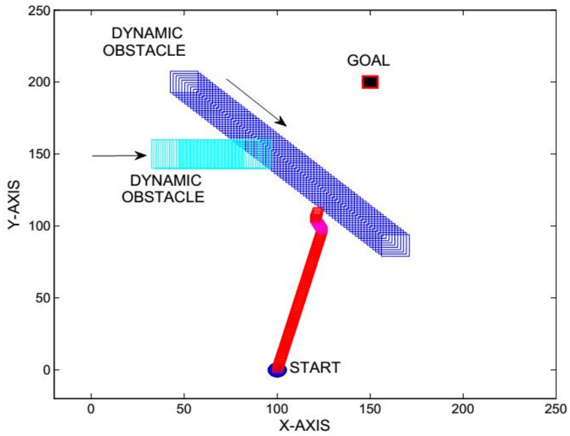


Figure 6.7 (b)

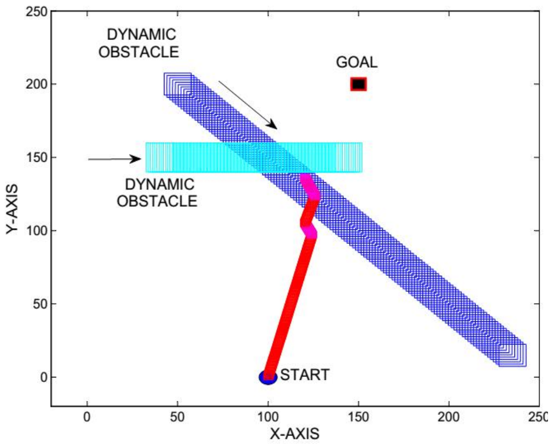


Figure 6.7 (c)

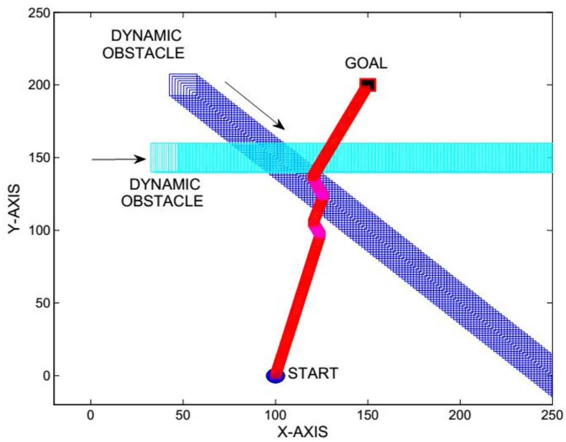


Figure 6.7 (d)

Figure 6.7: Navigation in presence of dynamic obstacles using FA controller

To understand the robot behavior during obstacle avoidance, the effect of variation in control parameter on performance of mobile robot navigation is checked by trial and error method. The Figure 6.8 shows the variation in path length due to change in a parameter such as N and β. During the test, $\gamma = 0.5$ and $\alpha = 0.5$ is considered.

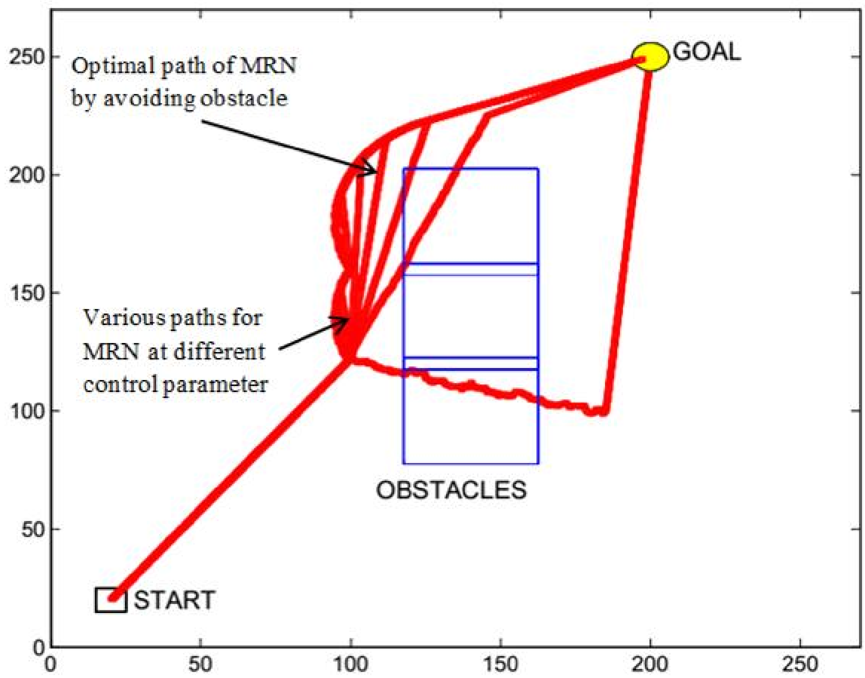


Figure 6.8: Navigation paths over different control parameter using FA controller

From the Table 6.2, after the many trials, it is clear that when the *N* is 50 and β is 0.2 then the path length is optimal and mobile robot avoids obstacle safely. The variation in *N* in range 40-60 and variation on β from 0.2 to 0.3 gives the nearby optimal solution and avoids the obstacle.

Table 6.2: Variation in path length due to change in control parameters

Sl. No.	*N*	β	Navigational path length in 'cm'	Obstacle avoidance (Yes/No)
1.	5	0.1	129.2	No
2.	15	0.15	105.6	No
3.	25	0.15	108.2	No
4.	**50**	**0.2**	**112.7(Optimal)**	**Yes**
5.	70	0.2	113.4	Yes
6.	90	0.25	114.1	Yes
7.	100	0.30	116.2	Yes

6.7 Experimental Analysis

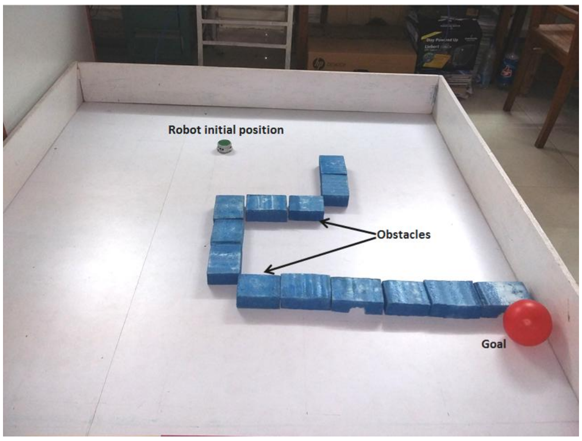


Figure 6.9 (a)

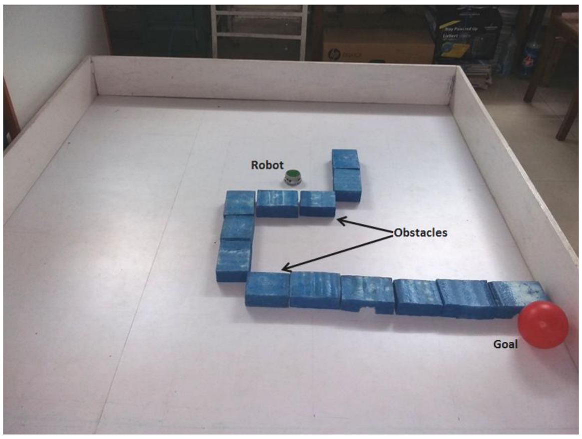


Figure 6.9 (b)

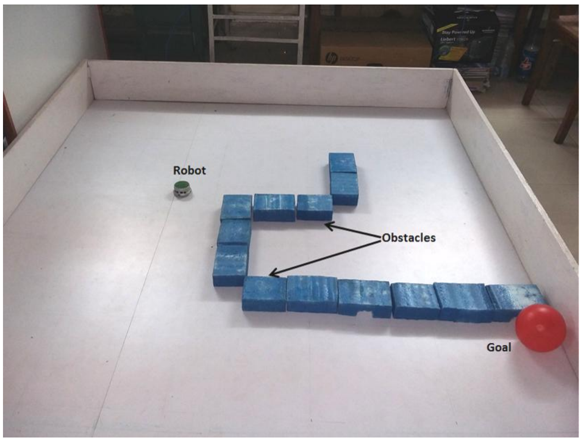


Figure 6.9 (c)

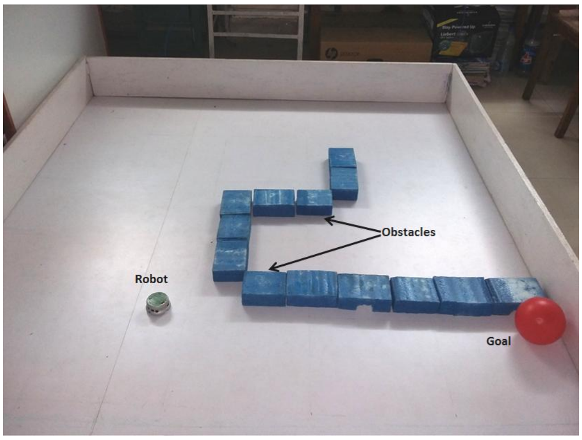


Figure 6.9 (d)

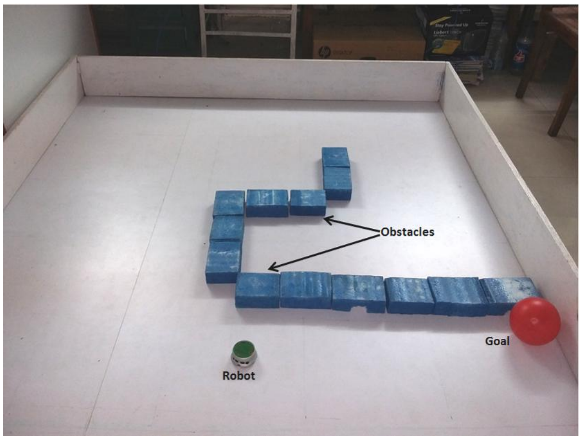


Figure 6.9 (e)

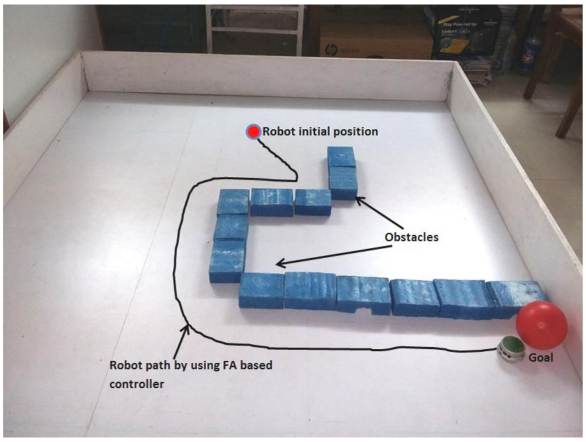


Figure 6.9 (f)

Figure 6.9: Real-time navigation using FA controller

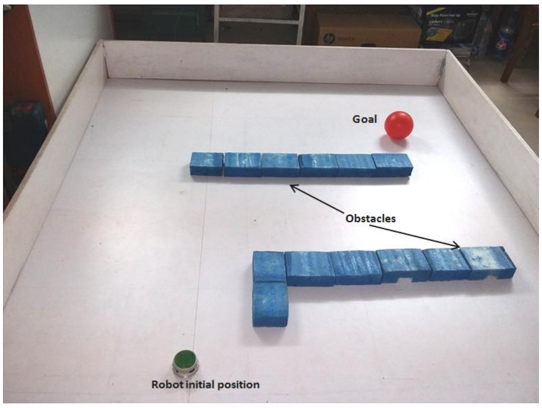


Figure 6.10 (a)

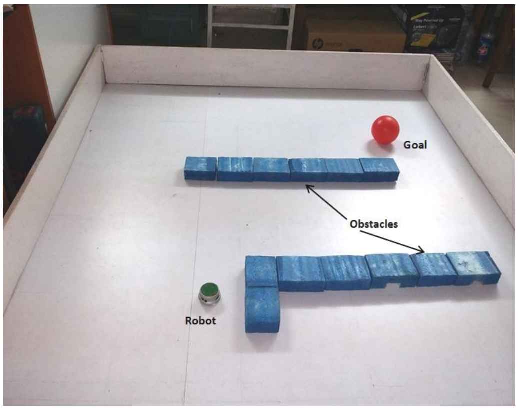


Figure 6.10 (b)

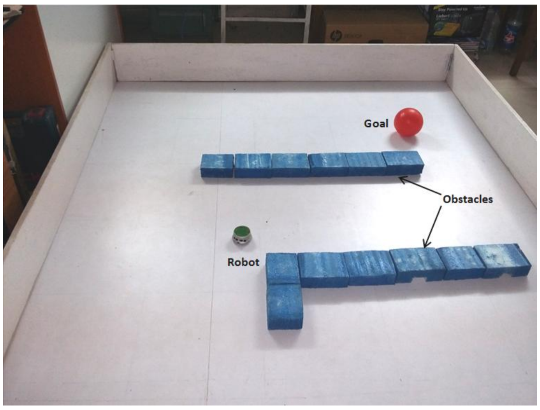


Figure 6.10 (c)

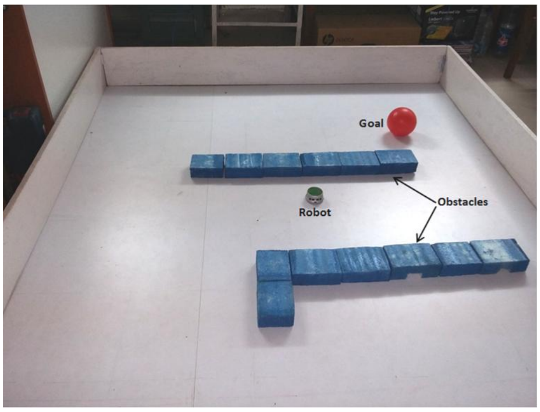


Figure 6.10 (d)

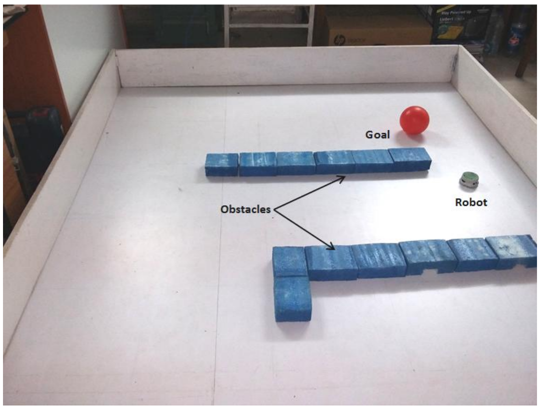


Figure 6.10 (e)

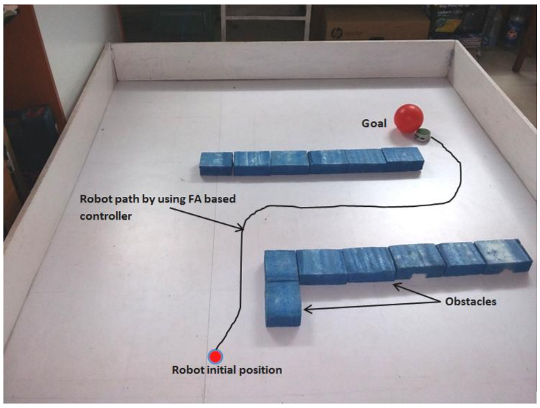


Figure 6.10 (f)

Figure 6.10: Real-time navigation using FA controller

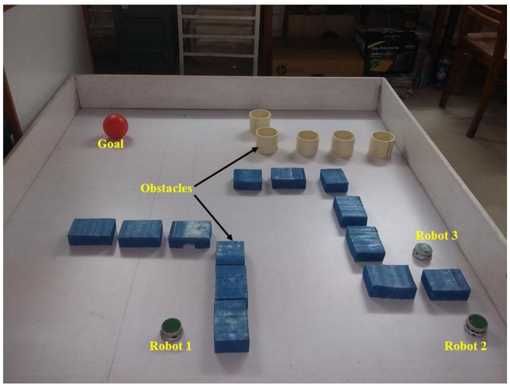


Figure 6.11 (a)

Figure 6.11 (b)

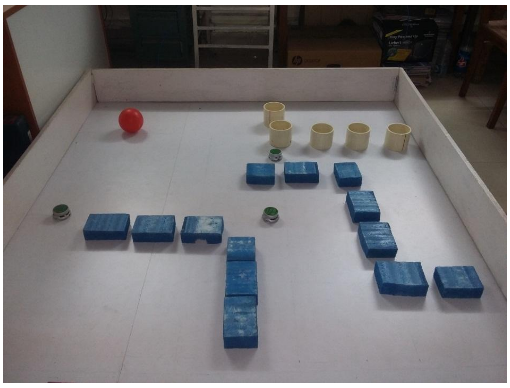


Figure 6.11 (c)

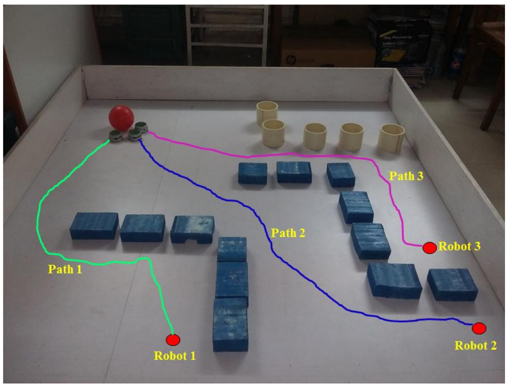


Figure 6.11 (d)

Figure 6.11: Real-time navigation using FA controller for multiple mobile robot

The feasibility of the FA based decision controller is demonstrated in a real-time environment in the presence of a variety of obstacles. For the navigational purpose, the Khepera-II robot is used. Khepera-II robot has eight infrared sensors which are capable of emitting and receiving the signals. These sensors are kept in a circular fashion around its body and sensors are allowed to measure distance in a short range of 1 cm to 5 cm. The technical specification of Khepera-II robot is given in Appendix. The C++ programming language is used for coding the program of MRN during the experiment. During the experiment (shown in Figure 6.2), it is clear that the robot directly reaches to goal when there are no obstacles present in the path of the robot and there is no need of FA based decision mechanism. The FA based decision mechanism activated when obstacles come in the path of the robot as shown in Figure 6.3.

6.8 Comparative Study of Experimental and Simulation Analysis of MRN over Similar Environment

To study the change in the functioning of the proposed FA based controller regarding path length and navigational time, the uniform environmental setup is considered for experimental and simulational analysis. The 20 trial runs are taken for comparative analysis for Scenario-1 (Figures 6.4 and 6.9) and Scenario-2 (Figures 6.5 and 6.10) whereas the 10 trials are taken for the multiple mobile robot system for Scenario-3 (Figure 6.6 and 6.11). The Table 6.3 and 6.4 gives path length comparision for Scenario 1 and Scenario 2 respectively whereas the Table 6.5 and 6.6 gives required navigational time for seenario 1 and Scenario 2 respectively for 20 trials. The Table 6.7 and 6.8 gives the comparision of path length and navigation time for Scenario 3 for 10 trials. It has been analyzed from the tabulation in Table 6.3-6.8 that the efficiency of the robot in the simulation is more than real time experiments. The calculated path length is less in the simulational analysis when compared to experimental analysis and percentage of deviation is within 5.4%. The required time of navigation while achieving the goal is more in the case of experimental analysis when compared to simulational analysis. The observed percentage of deviation is up to 6%.

Table 6.3: Path length in same simulational and experimental setup (Figure 6.4 and 6.9).

No. of runs	Experimental path length during MRN (in'cm')	Simulational path length during MRN (in 'cm')	% of deviation
1	191.00	180.20	5.61
2	191.32	181.00	5.39
3	191.22	181.30	5.18
4	191.89	181.15	5.59
5	191.94	181.11	5.64
6	191.88	181.80	5.25
7	191.41	181.65	5.09
8	191.33	180.45	5.68
9	191.68	180.60	5.78
10	191.79	180.25	6.10
11	192.11	182.00	5.26
12	192.00	181.85	5.28
13	191.23	180.50	5.61
14	191.42	180.10	5.91
15	191.15	181.90	4.83
16	191.57	182.10	4.94
17	191.76	180.47	5.36
18	191.50	180.79	5.59
19	190.87	181.40	4.96
20	190.95	181.37	5.01
Average path length covered	**191.50**	**181.14**	**5.40**

Table 6.4: Path length in same simulational and experimental setup (Figure 6.6 and 6.10).

No. of runs	Experimental path length during MRN (in'cm')	Simulational path length during MRN (in 'cm')	% of deviation
1	268.70	253.90	5.50
2	267.55	254.45	4.89
3	269.30	254.38	5.54
4	269.76	255.54	5.27
5	270.29	254.26	5.93
6	269.02	255.11	5.17
7	268.92	254.90	5.21
8	267.92	252.83	5.63
9	270.25	255.99	5.27
10	268.34	254.92	5.00
11	267.32	252.59	5.51
12	270.57	256.13	5.33
13	271.29	257.15	5.21
14	268.33	256.36	4.46
15	269.19	256.12	4.85
16	268.36	255.20	4.90
17	266.15	252.95	4.95
18	269.77	253.86	5.89
19	268.26	254.77	5.02
20	268.54	255.08	5.01
Average path length covered	**268.84**	**254.28**	**5.22**

Table 6.5: Navigational time in same simulational and experimental setup (Figure 6.5 and 6.9).

No. of runs	Experimental time during MRN (in 'sec')	Simulational time during MRN(in 'sec')	% of deviation
1	30.40	28.90	4.93
2	30.57	28.41	7.06
3	29.49	27.51	6.71
4	31.30	29.97	4.24
5	29.00	27.38	5.58
6	29.77	28.21	5.24
7	29.87	28.36	5.05
8	29.29	27.52	6.04
9	30.19	28.41	5.89
10	30.14	28.56	5.24
11	30.10	28.60	4.98
12	30.06	28.23	6.08
13	29.56	27.93	5.51
14	29.41	27.86	5.27
15	30.38	28.80	5.20
16	31.60	30.00	5.06
17	31.97	29.83	6.69
18	30.89	30.21	5.50
19	31.68	30.05	5.14
20	29.35	27.88	5.00
Average time required	**30.25**	**28.63**	**5.52**

Table 6.6: Navigational time in same simulational and experimental setup (Figure 6.6 and 6.10).

No. of runs	Experimental time during MRN (in 'sec')	Simulational time during MRN (in 'sec')	% of deviation
1	34.1	32.5	5.24
2	34.21	32.57	5.34
3	33.31	31.49	6.02
4	35.37	33.3	6.38
5	33.18	31.57	5.42
6	34.61	32.96	5.31
7	34.16	31.87	7.24
8	33.32	31.29	6.66
9	34.21	32.19	6.45
10	34.36	32.14	7
11	34.4	32.1	7.22
12	34.03	32.06	6.33
13	33.73	31.56	6.98
14	33.66	31.46	7.08
15	34.66	32.38	7.4
16	36	33.6	6.66
17	35.83	33.97	5.19
18	36.21	33.89	6.4
19	35.75	33.68	5.79
20	33.88	32.25	4.81
Average time required	**34.59**	**32.44**	**5.82**

Table 6.7: Path length in same simulational and experimental setup (Figure 6.7 and 6.11).

No. of runs	Robot No.	Experimental path length during MRN (in'cm')	Simulational path length during MRN (in 'cm')	% of deviation
1	Robot - 1	162	154.31	4.74
	Robot - 2	232.2	219.8	5.34
	Robot - 3	178.2	169.74	4.74
2	Robot - 1	165.21	155.97	5.59
	Robot - 2	230.12	221.32	3.82
	Robot - 3	178	169	5.05
3	Robot - 1	161.74	153.87	4.86
	Robot - 2	230	219	4.78
	Robot - 3	179.54	168.45	6.17
4	Robot - 1	163.84	154.74	5.55
	Robot - 2	234.77	222.22	5.34
	Robot - 3	177.33	170.74	3.71
5	Robot - 1	164.33	156.11	5.00
	Robot - 2	232.7	221.75	4.7
	Robot - 3	179.2	168.21	6.13
6	Robot - 1	162.48	155	4.6

	Robot - 2	235.1	222.97	5.15
	Robot - 3	180.11	170.41	5.38
7	Robot - 1	162.4	152.45	6.12
	Robot - 2	234.95	223.35	4.93
	Robot - 3	182.41	171.84	5.79
8	Robot - 1	163.87	155.82	4.91
	Robot - 2	235.9	221.05	6.29
	Robot - 3	181.9	172.9	4.94
9	Robot - 1	166	155.88	6.09
	Robot - 2	234.66	220.79	5.91
	Robot - 3	183.54	173.52	5.45
10	Robot - 1	161.68	155.3	3.94
	Robot - 2	231.84	221.2	4.58
	Robot - 3	179.63	171.58	4.48
Average path length covered	**Robot - 1**	**163.35**	**154.94**	**5.14**
	Robot - 2	**233.22**	**221.34**	**5.08**
	Robot - 3	**179.98**	**170.63**	**5.18**

Table 6.8: Navigational time in same simulational and experimental setup (Figure 6.7 and 6.11).

No. of runs	Robot No.	Experimental time during MRN (in 'sec')	Simulational time during MRN (in 'sec')	% of deviation
1	Robot - 1	28.06	26	7.34
	Robot - 2	38.34	36.3	5.32
	Robot - 3	29.6	28	5.40
2	Robot - 1	26.84	25.1	6.48
	Robot - 2	38.72	36.21	6.48
	Robot - 3	29.89	28.37	5.08
3	Robot - 1	25.41	24.6	3.18
	Robot - 2	39.1	37.05	5.24
	Robot - 3	28.97	28	3.34
4	Robot - 1	27.1	25.89	4.46
	Robot - 2	39.62	37.19	6.13
	Robot - 3	30.74	29.22	4.94
5	Robot - 1	26.12	25.24	3.36
	Robot - 2	37.1	35.73	3.69
	Robot - 3	31.74	29.67	6.52
6	Robot - 1	28.21	26.1	7.47

	Robot - 2	40.1	38.12	4.93
	Robot - 3	30.94	28.56	7.69
7	Robot - 1	27.47	25.4	7.53
	Robot - 2	40.75	38.45	5.64
	Robot - 3	29.99	28.83	3.86
8	Robot - 1	26.41	25.27	4.31
	Robot - 2	39.41	37.4	5.1
	Robot - 3	30.81	28.44	7.69
9	Robot - 1	26.87	24.9	7.33
	Robot - 2	38.74	36.27	6.37
	Robot - 3	31.37	29.36	6.40
10	Robot - 1	25.62	24.62	3.9
	Robot - 2	40.24	38.41	4.54
	Robot - 3	31	29.1	6.12
Average path length covered	**Robot - 1**	**26.81**	**25.31**	**5.54**
	Robot - 2	**39.21**	**37.11**	**5.34**
	Robot - 3	**30.50**	**28.75**	**5.70**

6.9 Performance Analysis of FA Controller with Other Navigational Controllers

In this section, the proposed FA based controller is compared with the other navigational controllers to check the optimality regarding path length. The graphical comparison is presented over similar simulation set up. The performance of proposed FA based controller is compared with the 2 stage neuro-fuzzy controller (Figure 6.12), and genetic algorithm (Figure 6.14) is shown in Table 6.9. It is clear that by using the FA controller we can save path length up to 6.06% when compared with the neuro-fuzzy approach provided by Joshi and Zaveri [200]. Similarly, the 6.54% path length can be saved by FA controller when compared to genetic algorithm by Wang et al. [201].

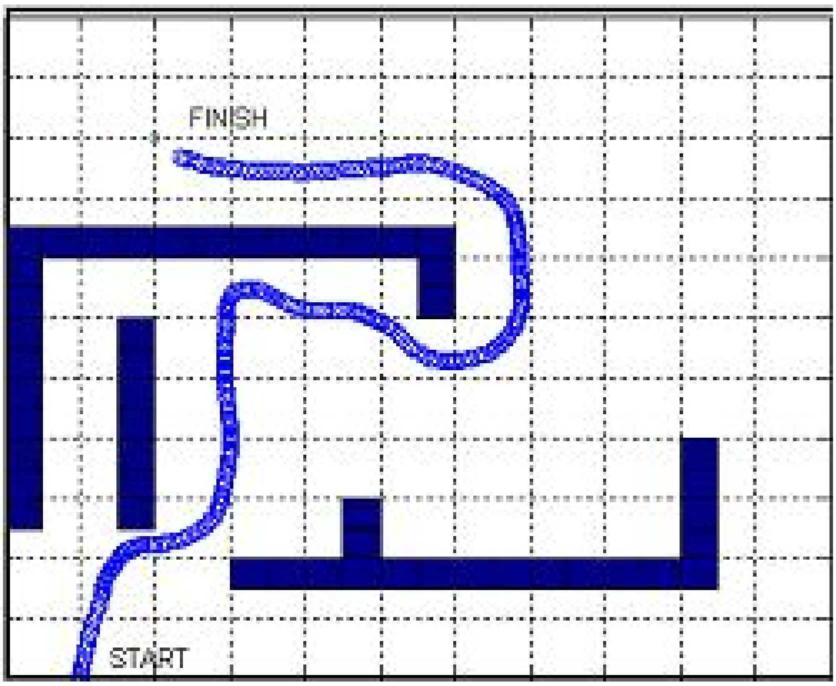


Figure 6.12: Navigation using neuro-fuzzy (Joshi & Zaveri [200])

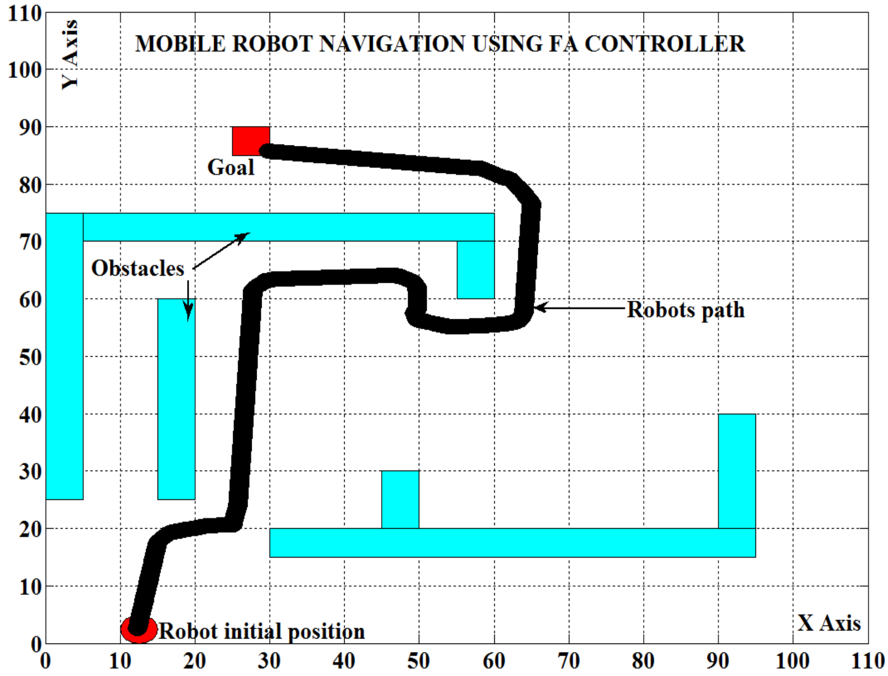


Figure 6.13: Navigation using FA controller

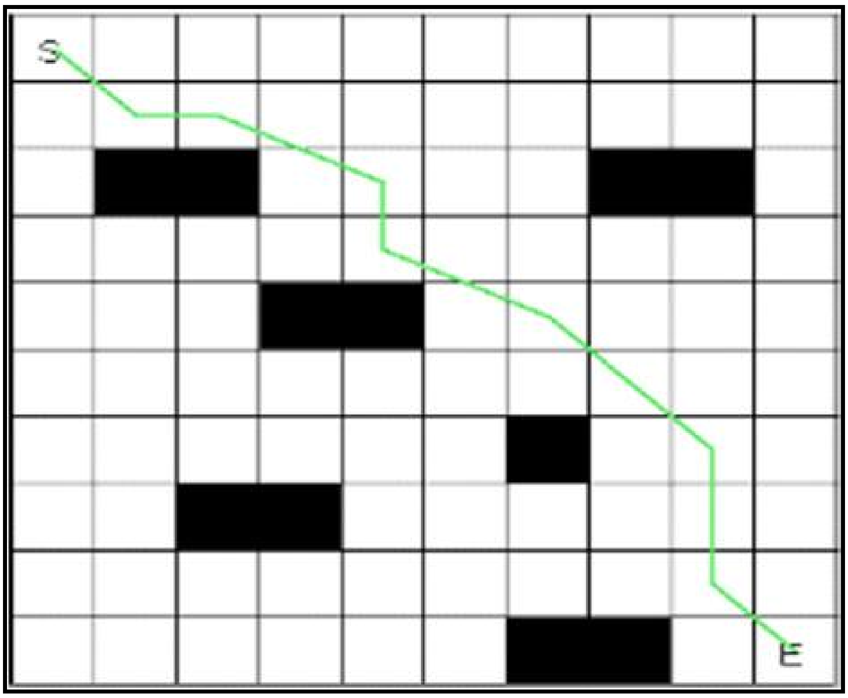


Figure 6.14: Navigation using genetic algorithm (Wang et al. [201])

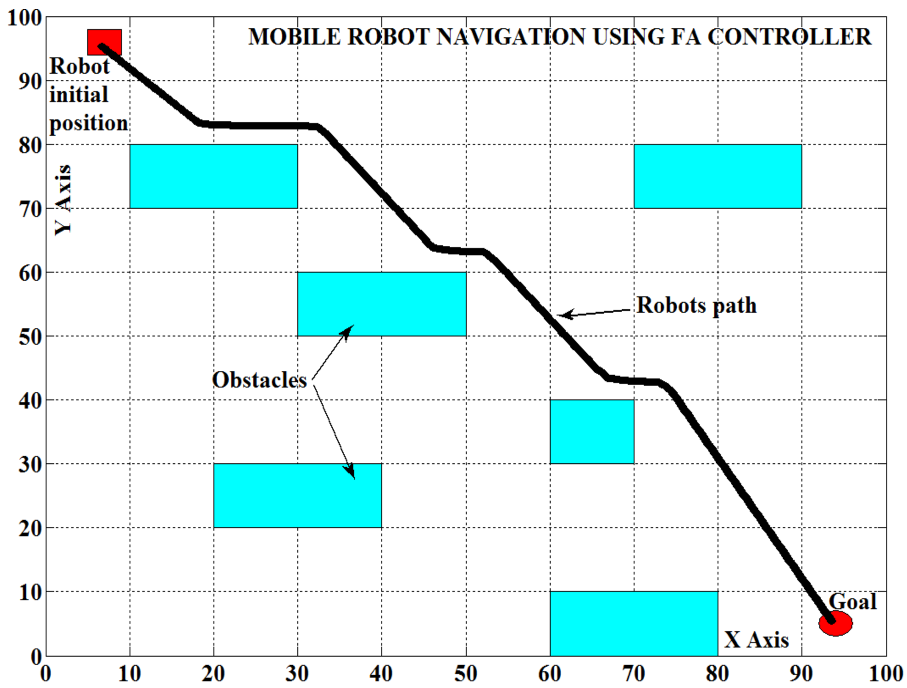


Figure 6.15: Navigation using FA controller

Table 6.9: Path length analysis of FA controller with other AI controller.

Sl. No.	Simulational path (in 'cm') by other AI controller	Simulational path (in 'cm') by FA controller	% of path saved by FA controller
Scenario-1	13.2 (Figure 6.12)	12.4 (Figure 6.13)	6.06
Scenario-2	10.7 (Figure 6.14)	10 (Figure 6.15)	6.54

6.10 Summary

This chapter gives the rigorous study of the FA based controller for navigation of mobile robot in the presence of static and dynamic obstacles. The followings are the key finding of the proposed work from the observed results as:

- The proposed FA based controller finds the brighter firefly among the group of fireflies in minimum time at low computational cost and considers it as a new position of the robot in the environment. This characteristic of present work generates effective fitness function for path planning of mobile robot navigation problem.

- The proposed controller successfully avoids the obstacles (static and dynamic) and also achieves the goal efficiently.
- The observed percentage of error for same experimental and simulational setup is less than 5.5% when compared in terms of path length and 6% when compared in terms of time taken for navigation.
- On comparison with other AI based controller (Neuro-Fuzzy and Genetic algorithm), it is observed that FA based controller saves the path length upto 6.54%.

At last, it is concluded that the proposed FA controller can be successfully applied to path planning problem in the uncertain environment.

Chapter 7

Hybrid Techniques for Mobile Robot Navigation

Now a day, much focus is given on the development of the hybrid algorithm for mobile robot navigation by many researchers. The aim of the current research is to design and develop the hybrid navigational controller by using firefly algorithm, probability-fuzzy logic and matrix based genetic algorithm for different environmental conditions.

7.1 Introduction

The approaches such as probability-fuzzy logic, matrix based genetic algorithm and firefly algorithm are used individually for mobile robot navigation problem. The various simulational and real-time experiments are carried out on different environmental conditions for the path optimisation. The obtained results show the effectiveness of the proposed technique regarding robot path planning. But, the proposed controller may not be optimal path planner for all configurations hence many researchers are working on the hybrid path planning approaches. In hybrid path planning approaches, more than one approaches are combined for hybridization to take the benefit of both. The key advantage of the hybrid approach is that it combines the multiple features of the different approaches into a single controller. The hybrid technique can be effectively used for optimal path planning of single and multiple wheeled mobile robots in an uncertain environment in the presence of static and dynamic obstacles. The chapter presents the hybrid of three different algorithms such as firefly algorithm, matrix based genetic algorithm and probability-fuzzy logic to validate the best combination for mobile robot navigation. The hybrid approaches such as Firefly Algorithm-Probability Fuzzy Logic (FA-PFL), Firefly Algorithm-Matrix based Genetic Algorithm (FA-MGA) and Firefly Algorithm- Probability-Fuzzy Logic-Matrix based Genetic Algorithm (FA-PFL-MGA) are used for navigation. In the current investigation, the output of the one particular algorithm is used as one of the inputs for another technique to hybridize the different techniques. The discussion of the mentioned

MGA, PFL and FA approaches is already presented in previous Chapter- 4, Chapter-5 and Chapter-6 respectively.

The section below presents the novel approach of using best things of the different algorithms to get better and effective results regarding acceptability and efficiency. Sometimes, the single intelligent approach may fail to give the required optimal solution. Hence, to overcome these difficulties hybridized algorithm has been developed. The objective of the hybrid controller is to improve the performance of the system in the presence of uncertainty, to handle the system with large data and to improve the convergence rate.

7.2 Application of FA for Hybridization

This section presents the application of the FA as a common initial controller for the development of the all the three hybrid controllers. While working on the fuzzy logic based controller and genetic algorithm based controllers, it has been observed that the training and updating of the premise and its parameter are difficult. The difficulty is not only about the training of the parameters but also the calculation of gradient in each step. These controllers use the gradient based algorithm for training the parameters which may reduce the performance of the system. The gradient descent method has some limitation such as difficulty to find the best fit parameter and poor convergence rate of the parameter. The limitation of the gradient descent method has been minimized by using the FA. The FA is used as the initial part of the all the three controllers to provide the best fit parameters to the next controller. The FA along with the least square estimation (LSE) is used to train the initial parameters of the premises and then to find the resulting optimal parameter. The FA provides the well-tuned parameter to the next controller to achieve the desired task by minimizing the errors. FA is nature based metaheuristic algorithm which is developed from the biological behavior of the fireflies hence it does not requires any training and achieves the faster conversion rate of parameters for another controller. This ability of FA has preferred it for the solving high-level optimization problem.

In the three proposed hybrid controllers, the firefly algorithm (FA) initially trains the premise part and LSE approach is used to train the resulting parameter of the other controller. All three hybrid controllers are modeled by considering the robot to obstacle distance, robot to goal distance, mutual distance between the robot and their motion. All navigation parameters are filtered in the controller to give required steering angle for the robot. The Petri-Net approaches avoid the collision among the robot. The real-time experimental analysis is done on a mobile robot which is explained in next section of the chapter. The simultaneous comparison is made between the simulation and real-time environmental result to validate the accountability of the proposed controller in various environmental Scenarios. To check the feasibility of the proposed hybrid controllers, the results are compared with the different intelligent approaches.

7.3 Analysis of FA-PFL Hybrid Controller for Navigation

The section presents FA-PFL hybrid navigational controller by combining the firefly algorithm (FA) and probability-fuzzy logic (PFL). The PFL and FA controller are explained previously in Chapter 4 and Chapter 6 respectively. The proposed hybrid optimal path planner is developed based on a reference motion, direction, distances between the robot to obstacles and distance between the robot to goal. In order to avoid a collision against each other, a set of collision prevention rules are embedded into each robot controller, using Petri-Net model.

Figure 7.2 presents the architecture of the FA-PFL hybrid controller to calculate the desired heading angle for optimal path planning in the robot environment. In the first part of the FA-PFL controller, the inputs to the FA controller are FOD, LOD and ROD whereas the output is intermediate heading angle (IHA). The output of the first FA controller i.e. IHA is the input for the second PFL controller. Apart from the IHA, the FOD, ROD and LOD with respect to robot existing position are also the input for the PFL controller. To find the FOD, ROD and LOD, the robot is equipped with eight IR sensors around its periphery and it also detects the position of the goal. The hybrid controller considers the input from the FA controller and the respective distances from the obstacle for giving the output IHA. The obtained output from the FA controller is used to train the PFL to get the final heading angle (FHA) for all environmental condition. To analyze the robot performance in the real time environment, the program is embedded in the robot microcontroller.

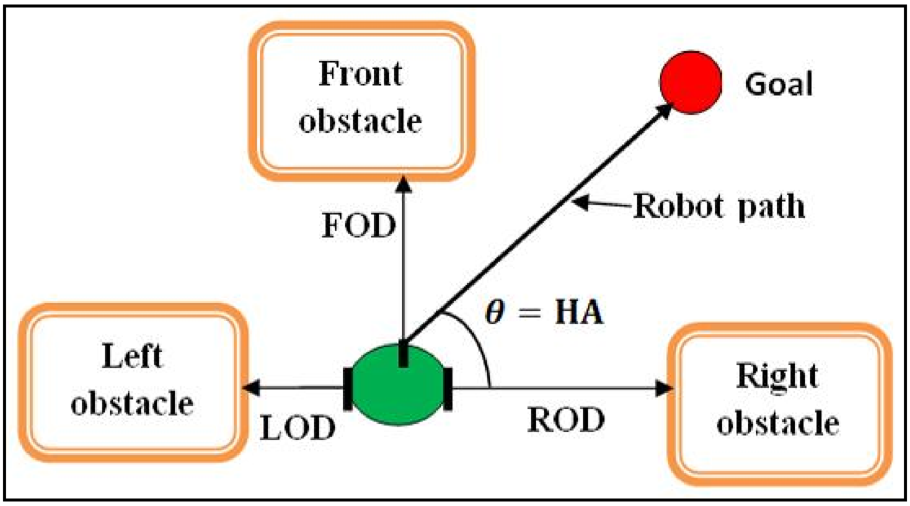


Figure 7.1: Robot position in environment with respect to obstacle

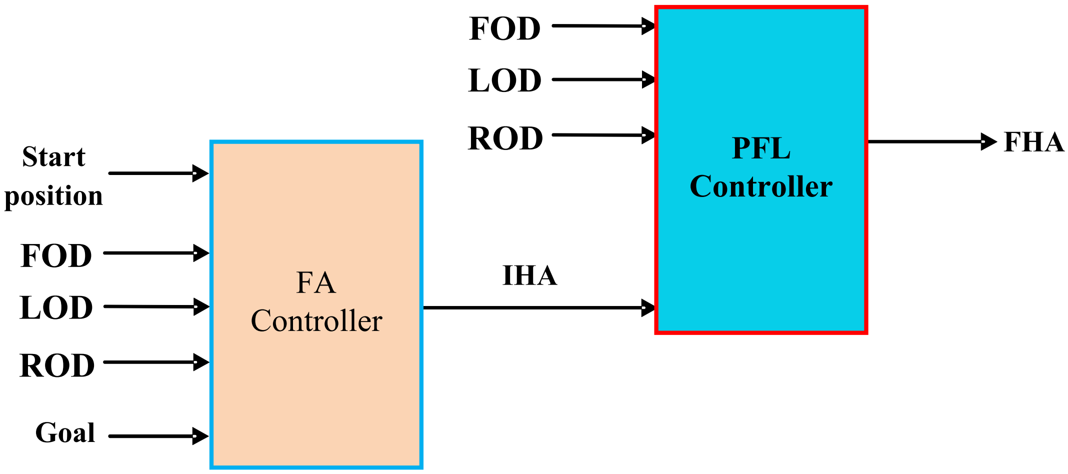


Figure 7.2: Hybrid FA-PFL controller for navigation

The simulation and experimental results for single and multiple robots using FA-PFL hybrid controller are presented in section 7.6.1 and comparison with the other AI controller is given in section 7.8.

7.4 Analysis of FA-MGA Hybrid Controller for Navigation

This section provides the new hybrid controller FA-MGA based on the firefly algorithm (FA) and matrix based genetic algorithm (MGA) for path planning of mobile robot. The proposed FA-MGA controller requires the initial information of the environment i.e. the obstacle distances as input to the FA controller. The FA controller gives the output as IHA and it becomes the input for MGA algorithm. Along with the IHA, the distance between the robot and obstacle such as FOD, LOD, ROD are also considered as the input to the MGA controller and this input data is to be trained in the MGA controller to get the final output FHA for navigation of wheeled mobile robot. The idea about the FA and MGA controllers is already presented in Chapter 6 and Chapter 5 respectively.

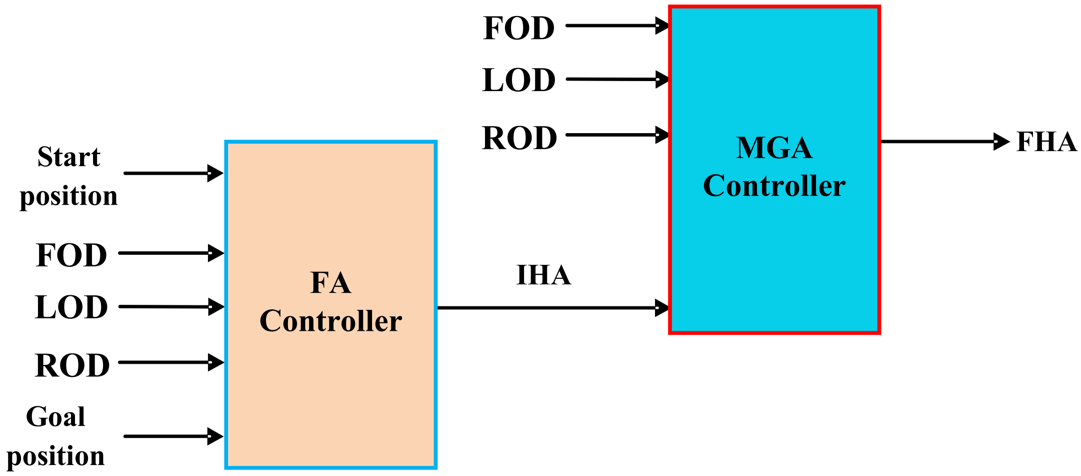


Figure 7.3: Hybrid FA-MGA controller for navigation

The simulation and experimental results for single and multiple wheeled robots using FA-MGA hybrid controller are presented in section 7.6.2 and comparison with the other AI controller is presented in section 7.8.

7.5 Analysis of FA-PFL-MGA Hybrid Controller for Navigation

This section presents the FA-PFL-MGA hybrid controller based on firefly algorithm, probability-fuzzy logic and matrix based genetic algorithm. The architecture of the proposed controller is presented in Figure 7.4. The execution of the algorithm such as MGA, PFL and FA is already explained in detail in the previous chapter. In the proposed hybrid controller, the initial input is given to the FA controller in the form of distances (FOD, LOD and ROD) then it gives the output in the form of first heading angle (FHA). The FHA, FOD, ROD and LOD are the inputs of the PFL controller in the next level which on training gives the output as second heading angle (SHA). The output from the PFC controller i.e. SHA along with FOD, ROD and LOD are the inputs for MGA controller which on training gives the desired heading angle (DHA) for navigation.

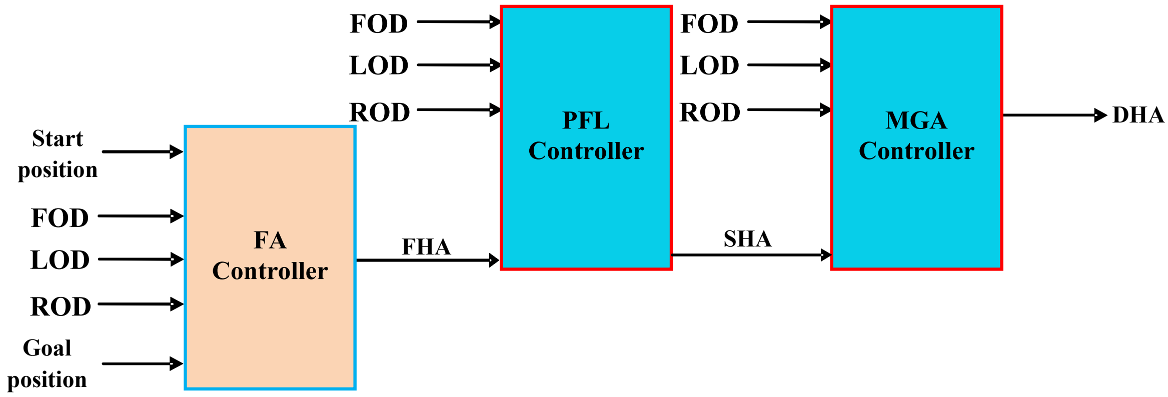


Figure 7.4: Hybrid FA-PFL-MGA controller for navigation

The simulation and experimental results for single and multiple wheeled robots using FA-PFL-MGA hybrid controller are presented in section 7.6.3 and comparison with the other AI controller is given section 7.8.

7.6 Simulational and Experimental Analysis of Hybrid Controller

In this section, the three different hybrid controller such as FA-PFL, FA-MGA and FA-PFL-MGA controllers are analyzed in the same simulational and experimental environment using Matlab software. The analysis has been performed in the presence of a static and dynamic obstacles. The navigation of single and multiple mobile robots is presented for the navigational task.

7.6.1 Simulational and Experimental Analysis of the FA-PFL Hybrid Controller

In this section, the simulational and experimental analysis of the FA-PFL hybrid controller has been examined for different environmental condition. To show the capability of the proposed controller, the various exercises have been conducted on single and multiple mobile robots in the static and dynamic environments. During analysis of the FA-PFL algorithm, the best parametric value of the firefly algorithm is considered and this is already discussed in Chapter 6.

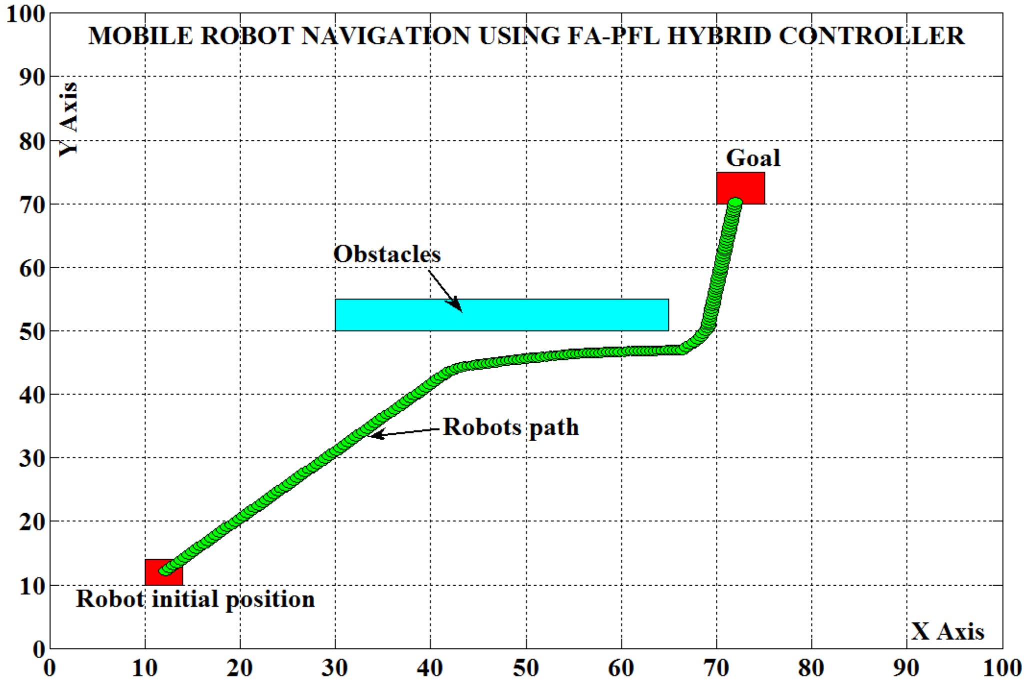


Figure 7.5: Navigation using FA-PFL hybrid controller

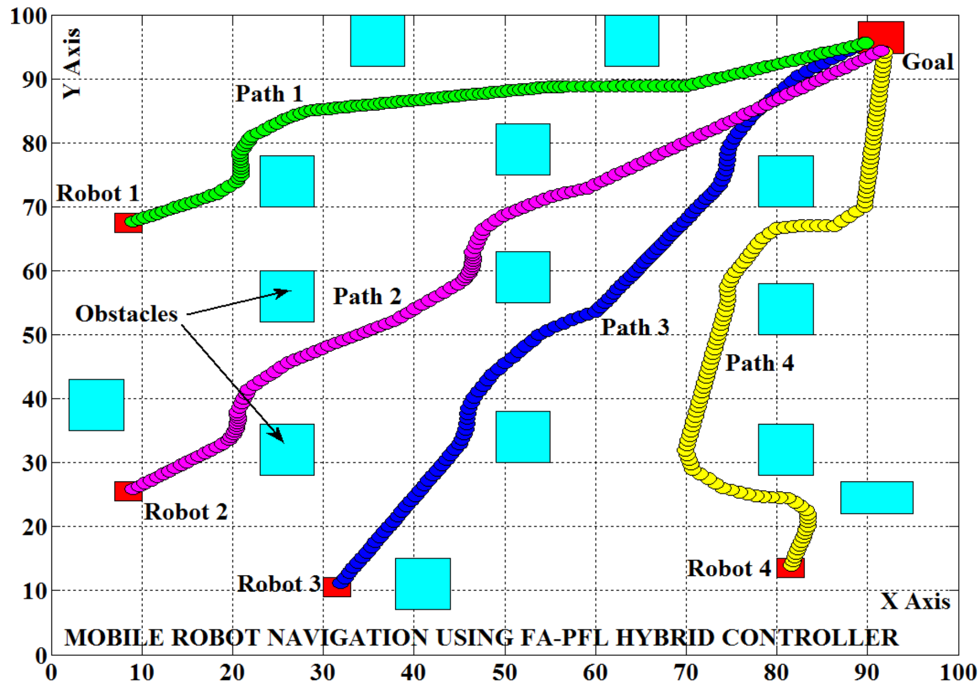


Figure 7.6: Navigation of multiple mobile robot using FA-PFL hybrid controller

The simulational analysis of the FA-PFL hybrid controller in the presence of moving obstacle is presented in Figure 7.7. The environment with one moving obstacle is presented to show the effectiveness in a dynamic environment. The Figure shows the capability of the robot to avoid the obstacles while reaching the goal.

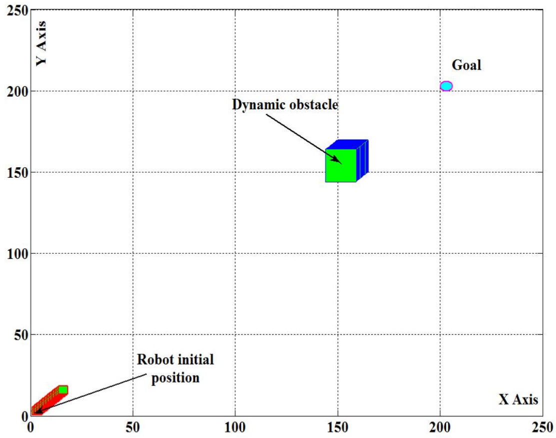


Figure 7.7 (a)

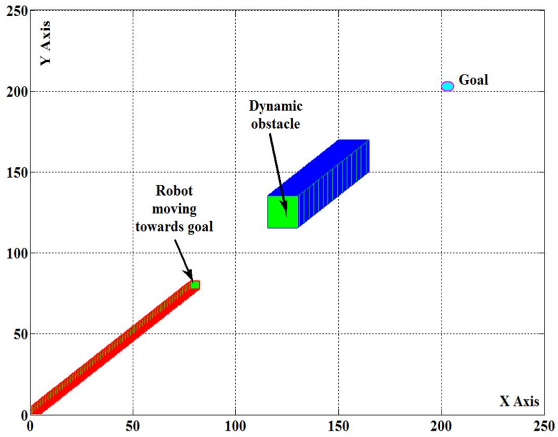


Figure 7.7 (b)

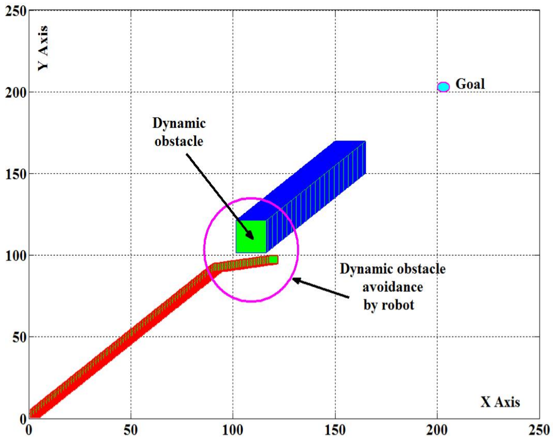


Figure 7.7 (c)

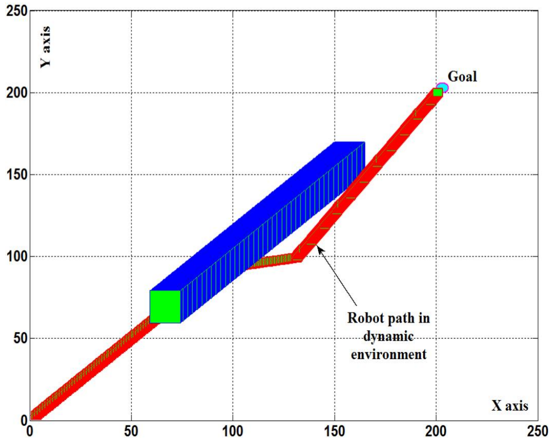


Figure 7.7 (d)

Figure 7.7: Navigation in dynamic environment using FA-PFL hybrid controller

The Figure 7.8 and 7.9 presents the real-time navigation of single and multiple WMR navigations. The real-time environment is created similar to the simulational environment for performance comparison.

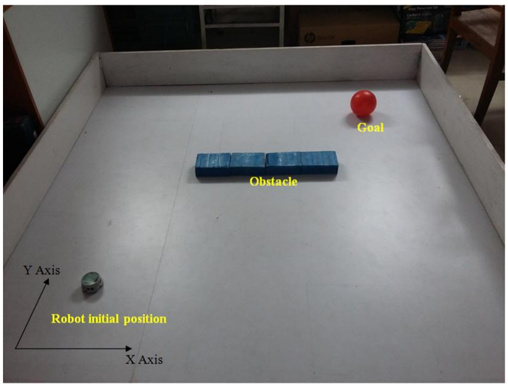


Figure 7.8 (a)

Figure 7.8 (b)

Figure 7.8 (c)

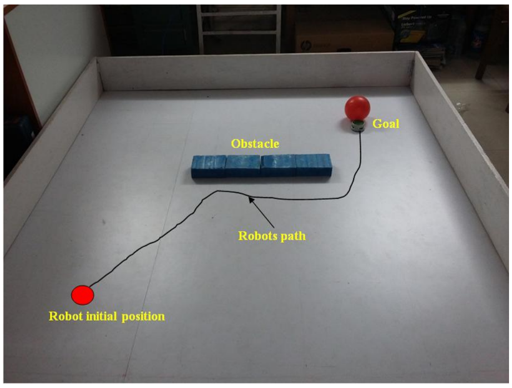


Figure 7.8 (d)

Figure 7.8: Real-time navigation using FA-PFL hybrid controller

Figure 7.9 shows the navigation of multiple mobile robot systems in presence and many obstacle [Dense environment]. The real time experiment is performed on the four Khepera-II robots which are located initially at various position. The final destination is same for all the robot.

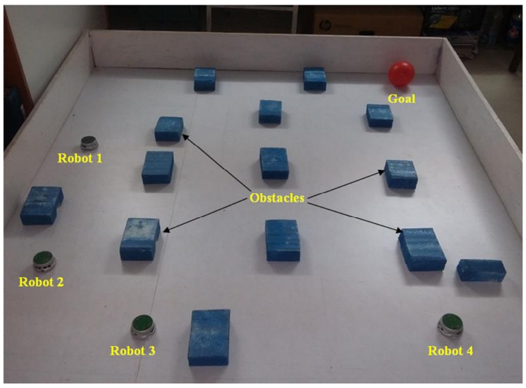


Figure 7.9 (a)

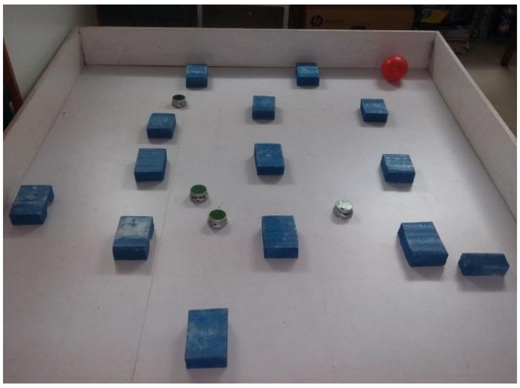


Figure 7.9 (b)

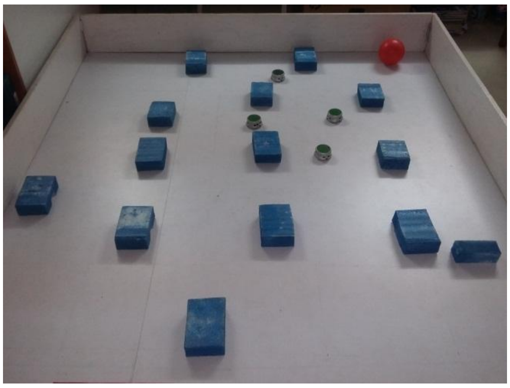


Figure 7.9 (c)

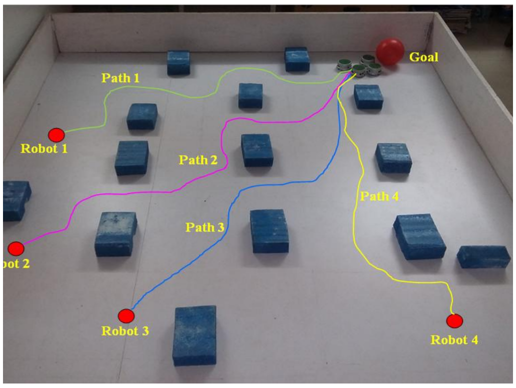


Figure 7.9 (d)

Figure 7.9: Real-time navigation using FA-PFL hybrid controller

7.6.2 Simulational and Experimental Analysis of the FA-MGA Hybrid Controller

In this section, the simulational and experimental analysis of the FA-MGA hybrid controller has been examined for different environmental condition. To show the capability of the proposed controller, various exercises have been conducted on the system of single and multiple mobile robots in the static and dynamic environment. During analysis of the FA-MGA controller, the best parametric value of the firefly algorithm is considered and it is already discussed in Chapter 6.

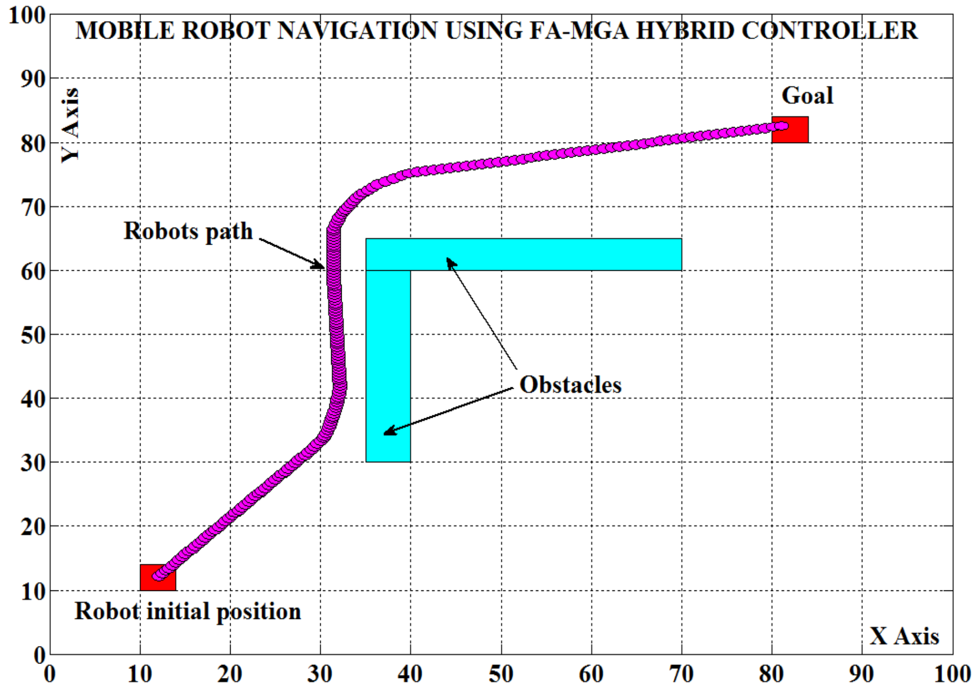


Figure 7.10: Navigation using FA-MGA hybrid controller

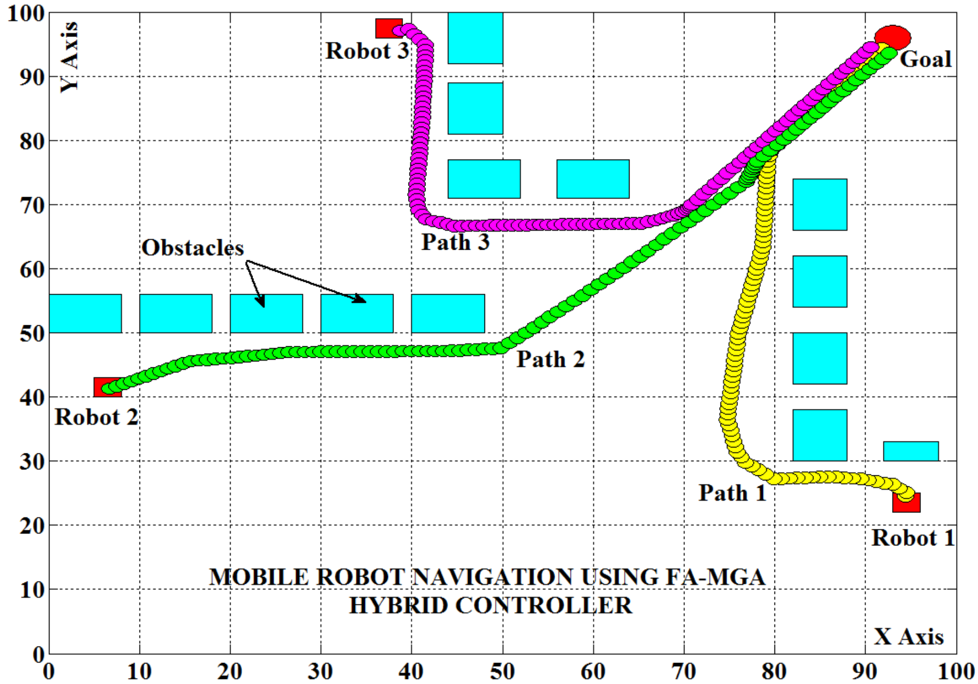


Figure 7.11: Navigation using FA-MGA hybrid controller

The simulational analysis of the FA-MGA hybrid controller in the presence of moving obstacle is presented in Figure 7.12. The environment with one moving obstacle is presented here to show the effectiveness in dynamic environment. The following Figures presents the effectiveness of the robot in dynamic environment while avoiding the obstacle.

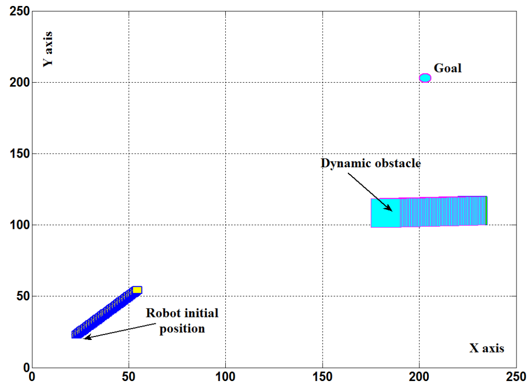


Figure 7.12 (a)

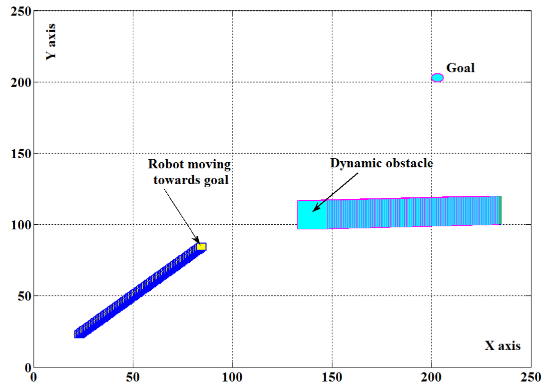


Figure 7.12 (b)

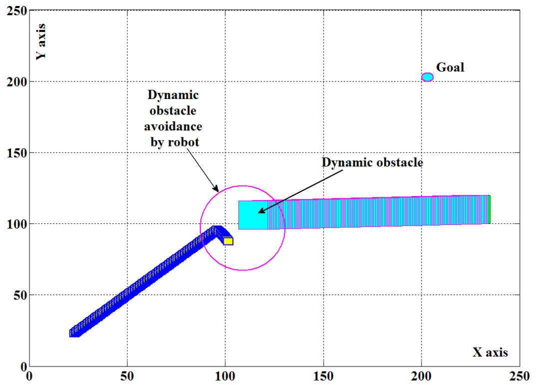


Figure 7.12 (c)

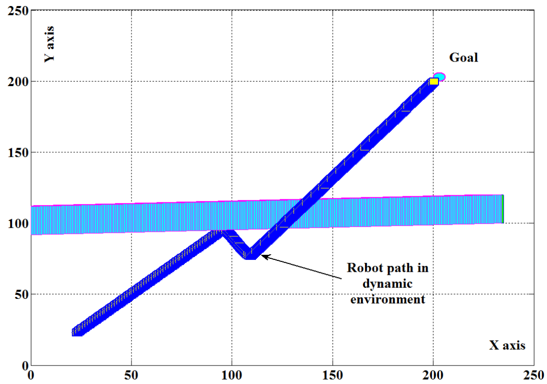


Figure 7.12 (d)

Figure 7.12: Navigation in dynamic environment using FA-MGA hybrid controller

The Figures 7.13 and 7.14 present the real-time navigation of single and multiple wheeled mobile robot in various environment condition. The real-time environment is created similar to simulational environment for performance analysis.

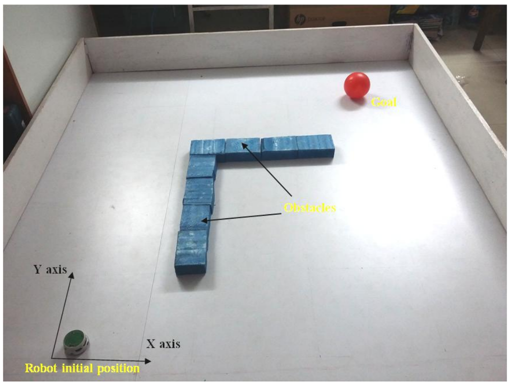


Figure 7.13 (a)

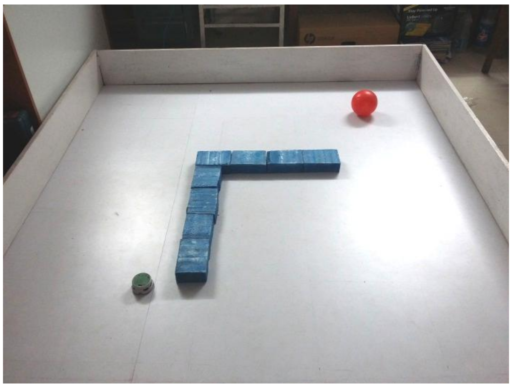


Figure 7.13 (b)

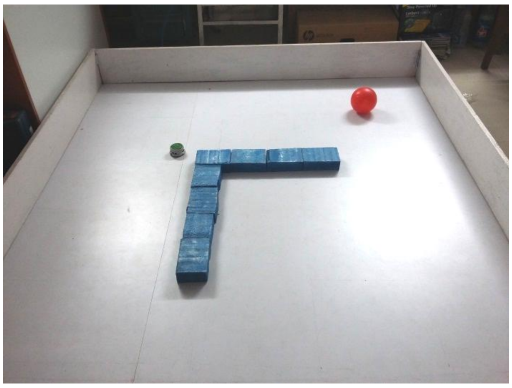


Figure 7.13 (c)

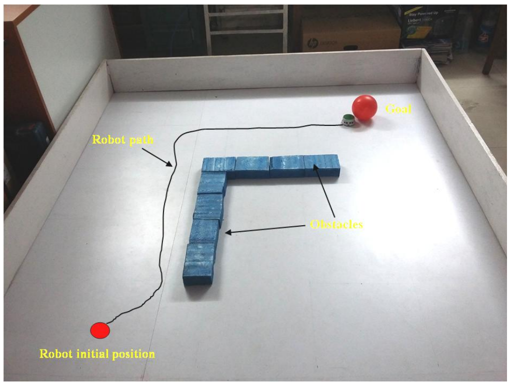


Figure 7.13 (d)

Figure 7.13: Real-time navigation using FA-MGA hybrid controller

The Figure 7.14 shows the navigation of multiple mobile robot systems in the clultered environment. The real time experiment is performed with the three Khepera-II robots which are located initially at various position. The final destination is same for all the mobile robot during navigation.

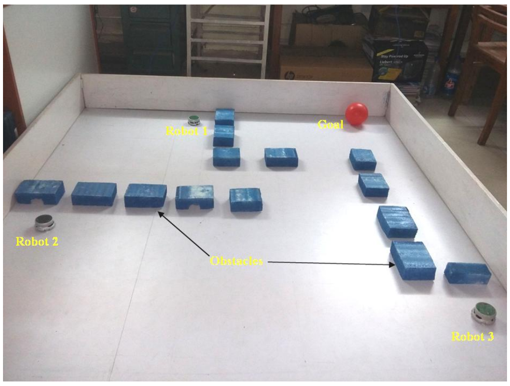


Figure 7.14 (a)

Figure 7.14 (b)

Figure 7.14 (c)

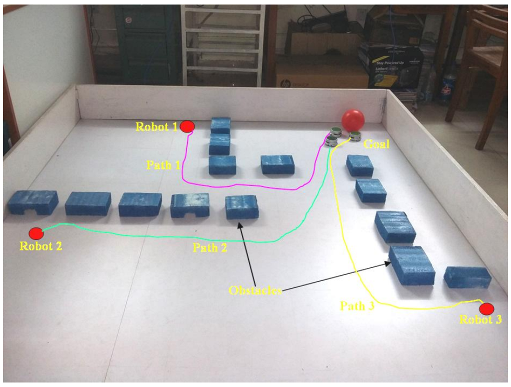


Figure 7.14 (d)

Figure 7.14: Real-time navigation using FA-MGA hybrid controller

7.6.3 Simulational and Experimental Analysis of the FA-PFL-MGA Hybrid Controller

In this section, the simulational and experimental analysis of the FA-PFL-MGA hybrid controller have been given for different environment condition. To show the strength of the proposed controller, the various exercises have been conducted on the system of single and multiple mobile robots in the static and dynamic environments. During analysis of the FA-PFL-MGA controller, the best parametric value of the firefly algorithm is considered and this is already discussed in Chapter 6.

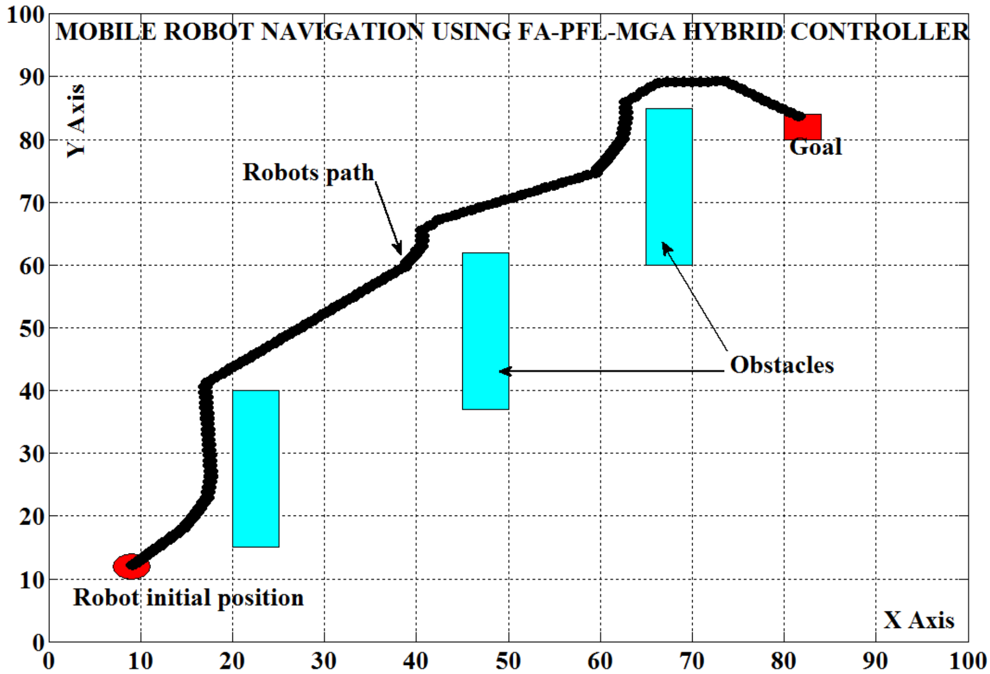


Figure 7.15: Navigation using FA-PFL-MGA hybrid controller

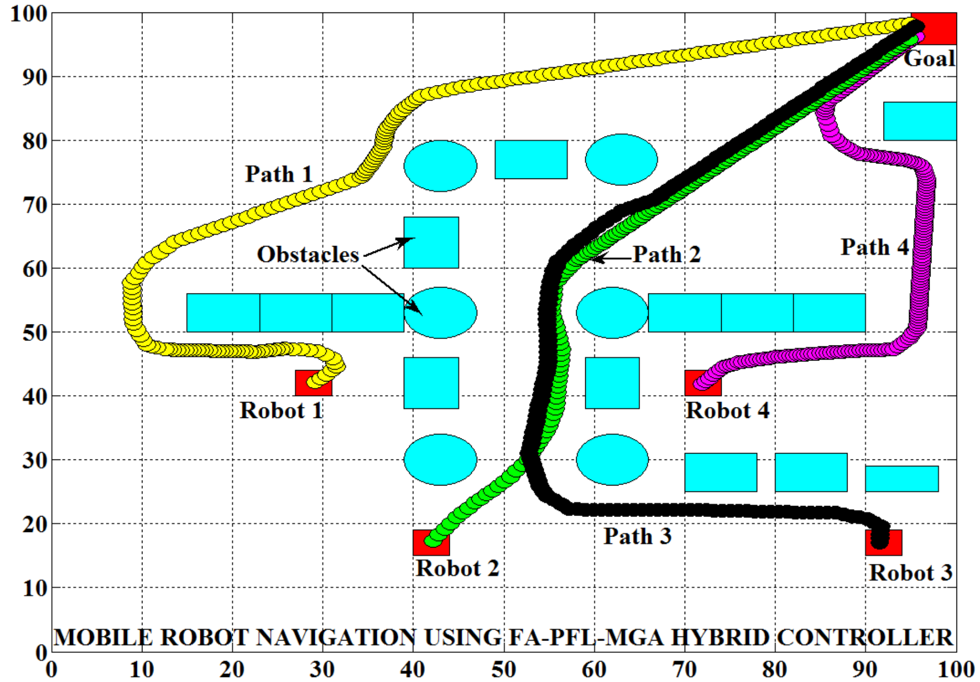


Figure 7.16: Navigation using FA-PFL-MGA hybrid controller

The simulational analysis of the FA-PFL-MGA hybrid controller in the presence of moving obstacle is presented in Figure 7.17. The environment with two moving obstacle is presented to show the effectiveness in a dynamic environment. The following Figures shows the robot is effectively avoiding the obstacle while reaching the goal.

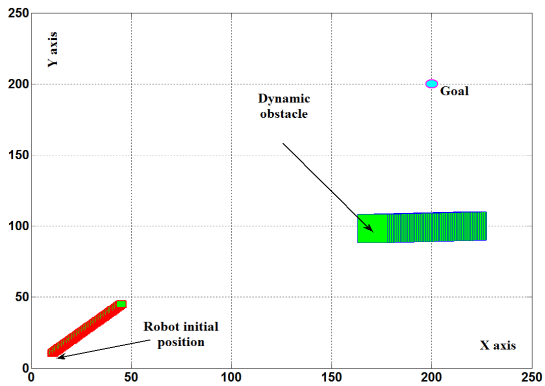


Figure 7.17 (a)

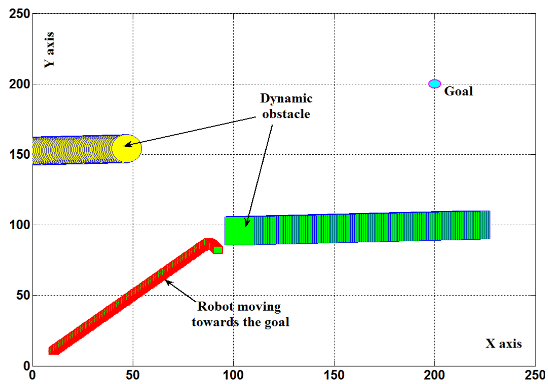


Figure 7.17 (b)

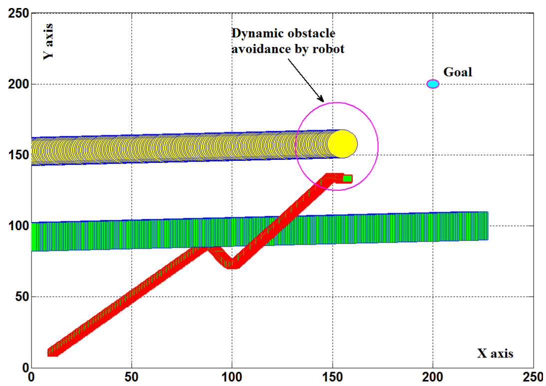


Figure 7.17 (c)

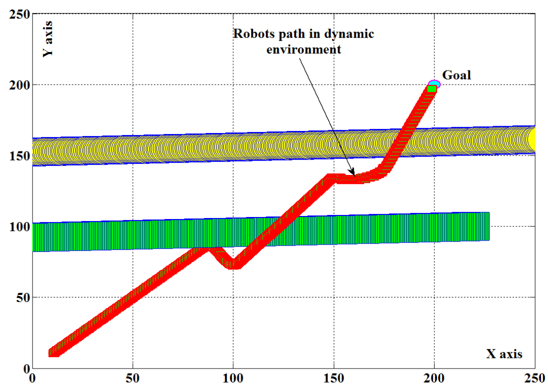


Figure 7.17 (d)

Figure 7.17: Navigation in dynamic environment using FA-PFL-MGA hybrid controller

The Figure 7.18 and 7.19 presents the real-time navigation of single and multiple wheeled mobile robot in various environment. The real-time environment is created similar to simulational environment for performance analysis.

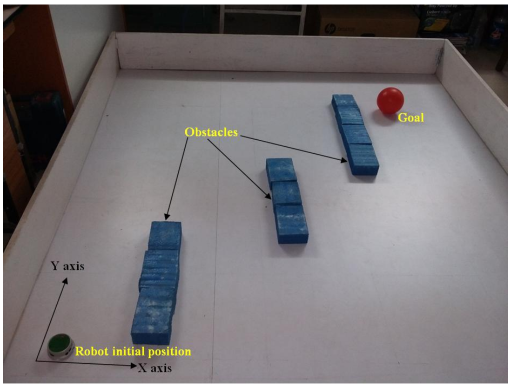


Figure 7.18 (a)

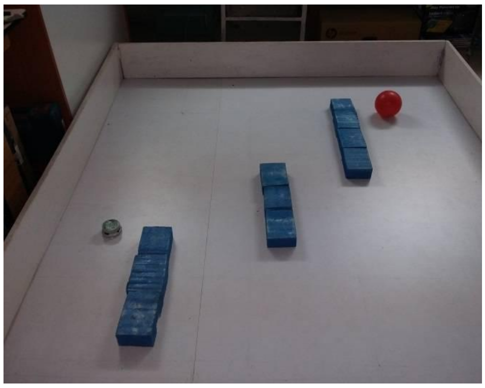


Figure 7.18 (b)

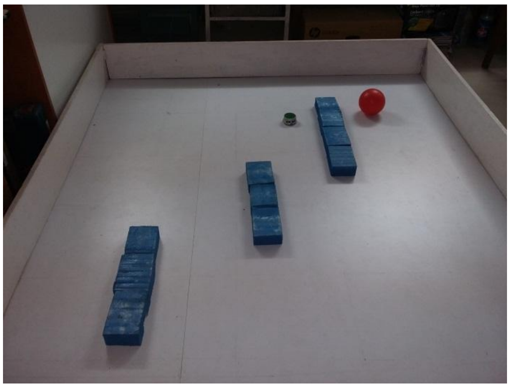


Figure 7.18 (c)

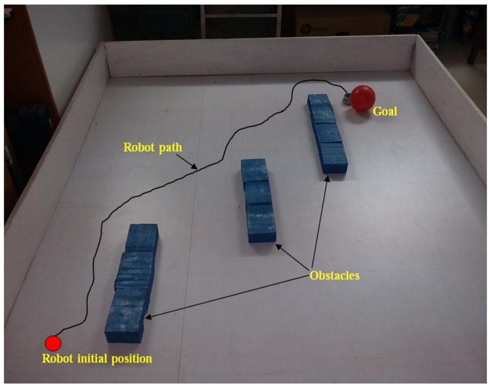


Figure 7.18 (d)

Figure 7.18: Real-time navigation using FA-PFL-MGA hybrid controller

The Figure 7.19 shows the navigation of multiple mobile robot systems in the cluttered environment. The real time experiment is performed on the four Khepera-II robots which are located initially at various position. The final destination is same for all the robot during navigation.

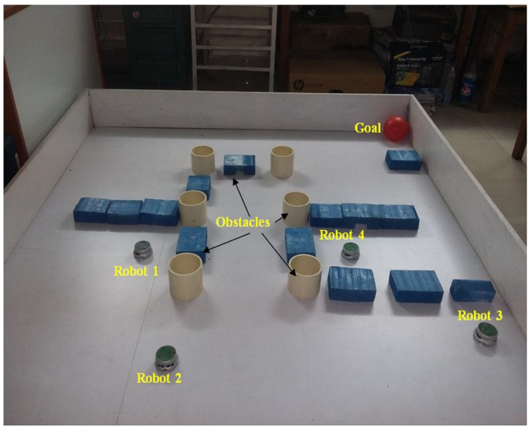


Figure 7.19 (a)

Figure 7.19 (b)

Figure 7.19 (c)

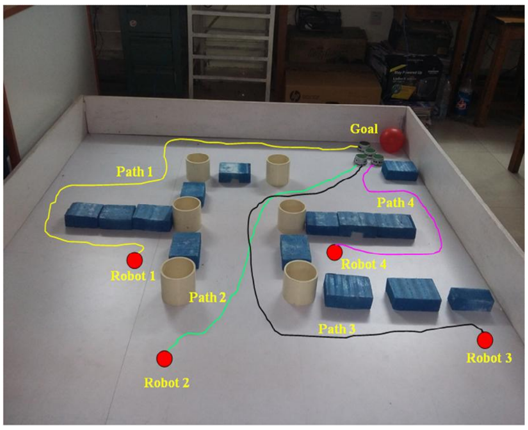


Figure 7.19 (d)

Figure 7.19: Real-time navigation using FA-PFL-MGA hybrid controller

7.7 Experimental and Simulation Performance Analysis of MRN over Similar Environment

The comparison between the experimental and simulational analysis is carried over the similar environmental setup for validation of the efficiency of the proposed hybrid controller. To verify the performance regarding path length and navigational time of the robot during the simulation and the real-time experiment, 20 trials are taken for single robot system and ten trials for multiple robot system. The following table gives the percentage of deviation in path length and navigational time over the same scenario for the individual hybrid controller.

Table 7.1: Path length in the same experimental and simulational environment using **FA-PFL** hybrid controller (Figure 7.5 and 7.8).

No. of runs	Experimental path length during MRN (in‘cm’)	Simulational path length during MRN (in ‘cm’)	% of deviation
1	163.73	155.32	5.13
2	161.38	155.23	3.81
3	160.15	154.57	3.48
4	163.98	154.12	6.01
5	162.5	155.63	4.22
6	163.32	155.25	4.94
7	164.1	155.69	5.12
8	165.06	156.82	4.49
9	161.2	155.28	3.67
10	162.39	155.41	4.29
11	161.65	154.37	4.50
12	161.74	155.87	3.62
13	164.77	156.34	5.11
14	161.97	153.9	4.98
15	160.12	154.24	3.67
16	160.14	153.64	4.05
17	163.85	153.25	6.46
18	163.68	155.1	5.24
19	164.12	156.05	4.91
20	166.41	156.91	5.70
Average path length covered	**162.81**	**155.41**	**4.71**

Table 7.2: Navigational time in the same experimental and simulational environment using **FA-PFL** hybrid controller (Figure 7.5 and 7.8).

No. of runs	Experimental time during MRN (in 'sec')	Simulational time during MRN (in 'sec')	% of deviation
1	26.11	25.15	3.67
2	26.35	24.99	5.16
3	26.97	25.6	5.07
4	26.84	25.2	6.11
5	27.32	25.78	5.63
6	27.87	26.34	5.48
7	25.9	24.87	3.97
8	26.5	25.64	3.24
9	26.44	25.08	5.14
10	27.12	26	4.12
11	26.49	25.12	5.17
12	27.1	26.05	3.87
13	27.21	25.4	6.65
14	26	25.1	3.46
15	26.2	25.1	4.19
16	27.74	26.12	5.83
17	25.64	24.49	4.48
18	25.32	24.32	3.94
19	25.84	24.86	3.79
20	26.55	24.87	6.32
Average time required	**26.57**	**25.30**	**4.77**

Table 7.3: Path length in the same experimental and simulational environment using **FA-PFL** hybrid controller (Figure 7.6 and 7.9).

No. of runs	Robot No.	Experimental path length during MRN (in'cm')	Simulational path length during MRN (in 'cm')	% of deviation
1	Robot - 1	148.5	141.5	4.71
	Robot - 2	189.32	180.4	4.71
	Robot - 3	170	161.11	5.22
	Robot - 4	156.66	150.11	4.18
2	Robot - 1	148.15	140.68	5.04
	Robot - 2	190.12	179.36	5.65
	Robot - 3	171.29	163.33	4.64
	Robot - 4	157.29	152	3.36
3	Robot - 1	149.2	142.35	4.59
	Robot - 2	188.36	179.19	4.86
	Robot - 3	171.56	162.77	5.12
	Robot - 4	157.87	150.98	4.36
4	Robot - 1	148.9	143	3.96
	Robot - 2	190.77	177.84	6.77
	Robot - 3	169.65	162.11	4.44
	Robot - 4	158.09	151	4.48
5	Robot - 1	147.25	142.74	3.06
	Robot - 2	190.07	180.81	4.87
	Robot - 3	170.19	162.87	4.30
	Robot - 4	157.51	150.8	4.26
6	Robot - 1	148.9	140.54	5.61

	Robot - 2	187.89	178.2	5.15
	Robot - 3	172.96	161.2	6.79
	Robot - 4	159.67	151.92	4.85
7	Robot - 1	148.57	140.7	5.29
	Robot - 2	188	179.57	4.48
	Robot - 3	170.47	163.1	4.32
	Robot - 4	156.83	151.66	3.29
8	Robot - 1	147.55	141.81	3.89
	Robot - 2	187.42	180.32	3.78
	Robot - 3	171.22	162	5.38
	Robot - 4	160.05	151.87	5.11
9	Robot - 1	148.24	140.1	5.38
	Robot - 2	188.98	180	4.75
	Robot - 3	170.05	163.6	3.79
	Robot - 4	159.42	152.2	4.52
10	Robot - 1	148.08	140.37	5.30
	Robot - 2	189	180.4	4.55
	Robot - 3	171.24	163.37	4.59
	Robot - 4	158	150.63	4.66
Average path length covered	**Robot - 1**	**148.33**	**141.37**	**4.68**
	Robot - 2	**188.99**	**179.60**	**4.96**
	Robot - 3	**170.86**	**162.54**	**4.86**
	Robot - 4	**158.13**	**151.31**	**4.31**

Table 7.4: Navigational time in the same experimental and simulational environment using **FA-PFL** hybrid controller (Figure 7.6 and 7.9).

No. of runs	Robot No.	Experimental time during MRN (in 'sec')	Simulational time during MRN (in 'sec')	% of deviation
1	Robot - 1	23.93	22.95	4.09
	Robot - 2	30.46	29.18	4.20
	Robot - 3	28.55	27	5.42
	Robot - 4	28.54	27.1	5.40
2	Robot - 1	23.15	22.42	3.15
	Robot - 2	30.85	29.09	5.70
	Robot - 3	27.23	25.87	4.99
	Robot - 4	28.23	26.87	5.49
3	Robot - 1	22.82	21.8	4.46
	Robot - 2	30.56	29.14	4.64
	Robot - 3	27	26.13	3.22
	Robot - 4	28.37	26.5	6.59
4	Robot - 1	22.73	21.3	6.29
	Robot - 2	31.2	29.95	4.00
	Robot - 3	28.37	26.5	6.59
	Robot - 4	26.17	25.08	4.16
5	Robot - 1	22.44	21.1	5.97
	Robot - 2	31.91	30.25	5.20
	Robot - 3	26.17	25.08	4.16
	Robot - 4	26	25	3.84
6	Robot - 1	24.62	22.92	6.90

	Robot - 2	31.97	30.59	4.31
	Robot - 3	26	25	3.84
	Robot - 4	28.34	26.9	5.08
7	Robot - 1	23.81	22.89	3.86
	Robot - 2	30	28.9	3.66
	Robot - 3	28.34	26.9	5.08
	Robot - 4	27.21	25.44	6.50
8	Robot - 1	22.54	21.43	4.92
	Robot - 2	32	30.4	5
	Robot - 3	27.21	25.44	6.50
	Robot - 4	26.47	25.27	4.53
9	Robot - 1	22.15	21.27	3.97
	Robot - 2	32.05	30.45	4.99
	Robot - 3	26.47	25.27	4.53
	Robot - 4	27	26.13	3.22
10	Robot - 1	23.97	22.77	5.00
	Robot - 2	32.13	30	6.62
	Robot - 3	28	26	4.12
	Robot - 4	28.2	26.83	4.85
Average time required	**Robot - 1**	**23.21**	**22.08**	**4.86**
	Robot - 2	**31.40**	**29.79**	**4.83**
	Robot - 3	**31.40**	**29.79**	**4.83**
	Robot - 4	**27.35**	**26**	**4.92**

Table 7.5: Path length in the same experimental and simulational environment using **FA-MGA** hybrid controller (Figure 7.10 and 7.13).

No. of runs	Experimental path length during MRN (in‘cm’)	Simulational path length during MRN (in ‘cm’)	% of deviation
1	194.4	184.2	5.24
2	194.12	187.3	3.51
3	194.87	185.64	4.73
4	194.75	183.75	5.64
5	193	186.74	3.24
6	193.77	187.26	3.35
7	196.08	185.84	5.22
8	194.1	183.07	5.68
9	195.7	183.63	6.16
10	193.02	184.38	4.47
11	195.25	186.44	4.51
12	194.67	184.73	5.10
13	197.31	187.54	4.95
14	194.82	187.97	3.51
15	195.85	187.3	4.36
16	193.34	185.91	3.84
17	193.71	185.7	4.13
18	197.45	186.8	5.39
19	194.11	184.1	5.15
20	197	184.4	6.39
Average time required	**194.86**	**185.63**	**4.73**

Table 7.6: Navigational time in the experimental and simulational environment using **FA-MGA** hybrid controller (Figure 7.10 and 7.13).

No. of runs	Experimental time during MRN (in 'sec')	Simulational time during MRN	% of deviation
1	31.33	30.34	3.15
2	31	29.87	3.64
3	31.87	30.78	3.42
4	32.78	30.48	7.01
5	31.92	30.73	3.72
6	32.12	29.88	6.97
7	33.17	32.12	3.16
8	32.84	30.87	5.99
9	32.19	30	6.80
10	31.74	31	2.33
11	32.74	31.14	4.88
12	32.94	31.05	5.73
13	31.05	29.9	3.70
14	32.85	30.65	6.69
15	32.84	31.2	4.99
16	32.84	31.81	3.13
17	31.87	30.41	4.58
18	31.93	30.9	3.22
19	32.55	30.94	4.94
20	32.19	30.7	4.62
Average time required	**32.23**	**30.73**	**4.63**

Table 7.7: Path length in the same experimental and simulational environment using **FA-MGA** hybrid controller (Figure 7.11 and 7.14).

No. of runs	Robot No.	Experimental path length during MRN (in‘cm’)	Simulational path length during MRN (in ‘cm’)	% of deviation
1	Robot - 1	135	129	4.44
	Robot - 2	190.02	179	5.79
	Robot - 3	167.4	161	3.82
2	Robot - 1	136.8	129.37	5.43
	Robot - 2	190.32	181.23	4.77
	Robot - 3	168.55	159.32	5.47
3	Robot - 1	134.25	129.71	3.38
	Robot - 2	192	184.35	3.98
	Robot - 3	168.52	160.75	4.61
4	Robot - 1	135.04	129.07	4.42
	Robot - 2	189.8	183.14	3.50
	Robot - 3	164.95	159.32	3.41
5	Robot - 1	134.65	130.12	3.36
	Robot - 2	189.7	180.97	4.60
	Robot - 3	167	161.97	3.01
6	Robot - 1	138.85	129.38	6.82

	Robot - 2	192.7	183.74	4.64
	Robot - 3	169.05	159.25	5.79
7	Robot - 1	137.45	129.28	5.94
	Robot - 2	190.57	183.5	3.70
	Robot - 3	168.5	159.67	5.24
8	Robot - 1	134.25	130.04	3.13
	Robot - 2	189.25	181.99	3.83
	Robot - 3	168.75	160.81	4.70
9	Robot - 1	137.19	131	4.51
	Robot - 2	191.25	179.96	5.90
	Robot - 3	165.34	157.1	4.98
10	Robot - 1	136.05	130.45	4.11
	Robot - 2	193.03	180.31	6.58
	Robot - 3	166	158.5	4.51
Average path length covered	**Robot - 1**	**135.95**	**129.74**	**4.55**
	Robot - 2	**190.86**	**181.81**	**4.73**
	Robot - 3	**167.40**	**159.76**	**4.58**

Table 7.8: Navigational time in the same experimental and simulational environment using **FA-MGA** hybrid controller (Figure 7.11 and 7.14).

No. of runs	Robot No.	Experimental time during MRN (in 'sec')	Simulational time during MRN (in 'sec')	% of deviation
1	Robot - 1	21.87	21	3.97
	Robot - 2	30.9	29	6.14
	Robot - 3	26.54	25	5.80
2	Robot - 1	21.05	20.4	3.08
	Robot - 2	31.29	29.87	4.53
	Robot - 3	27.23	25.9	4.88
3	Robot - 1	21.85	20.87	4.48
	Robot - 2	30.27	29.1	3.86
	Robot - 3	27.84	26.8	3.73
4	Robot - 1	22.3	21.45	3.81
	Robot - 2	31.75	30.1	5.19
	Robot - 3	26.41	24.7	6.47
5	Robot - 1	22	20.78	5.54
	Robot - 2	32	31	3.12
	Robot - 3	26.21	25.41	3.05
6	Robot - 1	21.98	21.09	4.04

	Robot - 2	32.15	30.82	4.13
	Robot - 3	26.97	26.09	3.26
7	Robot - 1	22.69	21.04	7.27
	Robot - 2	31.97	30.07	5.94
	Robot - 3	27.2	25.88	4.85
8	Robot - 1	21.41	20.36	4.90
	Robot - 2	30.74	29.28	4.74
	Robot - 3	28.31	26.39	6.78
9	Robot - 1	21.23	20.28	4.47
	Robot - 2	32.05	31	3.27
	Robot - 3	28.1	27.1	3.55
10	Robot - 1	22.8	21.61	5.21
	Robot - 2	32.83	30.78	6.24
	Robot - 3	27.64	26.35	4.66
Average path length covered	**Robot - 1**	**21.91**	**20.88**	**4.68**
	Robot - 2	**31.59**	**30.10**	**4.72**
	Robot - 3	**27.24**	**25.96**	**4.70**

Table 7.9: Path length in the same experimental and simulational environment using **FA-PFL-MGA** hybrid controller (Figure 7.15 and 7.18).

No. of runs	Experimental path length during MRN (in‘cm’)	Simulational path length during MRN (in ‘cm’)	% of deviation
1	210.11	201	4.33
2	210.85	203.3	3.58
3	210.74	202.8	3.76
4	211.05	201.74	4.41
5	210	202.87	3.39
6	212.33	204.94	3.48
7	214	202.01	5.60
8	214	202.63	5.31
9	211.87	202.56	4.39
10	214.52	204.76	4.54
11	213.82	204.6	4.31
12	210.97	200.7	4.86
13	210.46	203	3.54
14	214.68	204.81	4.59
15	215.01	204.05	5.09
16	214.96	203.9	5.14
17	215.87	201.18	6.80
18	213.87	204.64	4.31
19	213.02	204.74	3.88
20	211.4	203.98	3.50
Average path length covered	**212.67**	**203.21**	**4.44**

Table 7.10: Navigational time in the same experimental and simulational environment using **FA-PFL-MGA** hybrid controller (Figure 7.15 and 7.18).

No. of runs	Experimental time during MRN (in 'sec')	Simulational time during MRN (in 'sec')	% of deviation
1	33.86	32	5.49
2	30.12	29	3.71
3	30.98	29.11	6.03
4	33.79	32.4	4.11
5	31.84	30.41	4.49
6	32.17	30.94	3.82
7	30.74	29.67	3.48
8	32.98	31.86	3.39
9	33.05	31.89	3.50
10	33.32	31	6.96
11	32.51	31.12	4.27
12	31.1	30.16	3.02
13	30.45	29.3	3.77
14	30.85	29.22	5.28
15	33	32.3	2.12
16	31.55	30.33	3.86
17	32.5	30.47	6.24
18	30.47	29.34	3.70
19	32.93	31	5.86
20	31	29.63	4.41
Average time required	**31.96**	**30.55**	**4.38**

Table 7.11: Path length in the same experimental and simulational environment using **FA-PFL-MGA** hybrid controller (Figure 7.16 and 7.19).

No. of runs	Robot No.	Experimental path length during MRN (in‘cm’)	Simulational path length during MRN (in ‘cm’)	% of deviation
1	Robot - 1	213.3	204	4.36
	Robot - 2	148.5	142.9	3.77
	Robot - 3	205.2	194	5.45
	Robot - 4	135	128.3	4.96
2	Robot - 1	215.5	205.61	4.58
	Robot - 2	148.2	142.38	3.92
	Robot - 3	204	196.74	3.55
	Robot - 4	135.9	129.7	4.56
3	Robot - 1	213	205	3.75
	Robot - 2	148.25	143	3.54
	Robot - 3	205.78	197.31	4.11
	Robot - 4	134.41	129.14	3.92
4	Robot - 1	214.74	204.97	4.54
	Robot - 2	151.51	144.85	4.39
	Robot - 3	205.79	196.38	4.57
	Robot - 4	136.05	128.3	5.69
5	Robot - 1	216.08	206.08	4.62
	Robot - 2	151.74	142.36	6.18
	Robot - 3	206.12	197.52	4.17
	Robot - 4	135.4	131.4	2.95
6	Robot - 1	213.64	205.44	3.83

	Robot - 2	152.61	144.7	5.18
	Robot - 3	204.07	197.1	3.41
	Robot - 4	136.99	129	5.83
7	Robot - 1	214.05	205	4.22
	Robot - 2	149.11	144	3.42
	Robot - 3	204	195.55	4.14
	Robot - 4	137.23	131.1	4.46
8	Robot - 1	214	205.14	4.14
	Robot - 2	148.4	142.75	3.80
	Robot - 3	205	197.62	3.6
	Robot - 4	136	131.07	3.62
9	Robot - 1	213.87	206	3.67
	Robot - 2	150.04	142.77	4.84
	Robot - 3	205.8	194.7	5.39
	Robot - 4	134.55	128.74	4.31
10	Robot - 1	215.9	203.75	5.62
	Robot - 2	151.67	143.65	5.28
	Robot - 3	207.9	195.6	5.91
	Robot - 4	134.6	129.33	3.91
Average path length covered	**Robot - 1**	**214.40**	**205.09**	**4.33**
	Robot - 2	**150.00**	**143.33**	**4.43**
	Robot - 3	**205.36**	**196.25**	**4.43**
	Robot - 4	**135.61**	**129.60**	**4.42**

Table 7.12: Navigational time in the same experimental and simulational environment using **FA-PFL-MGA** hybrid controller (Figure 7.16 and 7.19).

No. of runs	Robot No.	Experimental time during MRN (in 'sec')	Simulational time during MRN (in 'sec')	% of deviation
1	Robot - 1	34.38	33	4.01
	Robot - 2	23.93	23	3.88
	Robot - 3	33.08	31.3	5.38
	Robot - 4	21.76	21	3.49
2	Robot - 1	34.34	33	3.90
	Robot - 2	24.3	23.4	3.70
	Robot - 3	32	30.7	4.06
	Robot - 4	21.01	20.4	2.90
3	Robot - 1	35.41	33.3	5.95
	Robot - 2	26.9	25.69	4.49
	Robot - 3	32.21	31.12	3.38
	Robot - 4	22.35	21.37	4.38
4	Robot - 1	36.74	35.5	3.37
	Robot - 2	24.74	23.5	5.01
	Robot - 3	33.56	32.01	4.61
	Robot - 4	22.35	21.37	4.38
5	Robot - 1	35	33.6	4
	Robot - 2	24.95	24.25	2.80
	Robot - 3	34.52	33.35	3.38
	Robot - 4	22.54	21.98	2.48
6	Robot - 1	33.12	31.24	5.67

	Robot - 2	25.24	23.78	5.78
	Robot - 3	33.85	31.9	5.76
	Robot - 4	23.45	22.1	5.75
7	Robot - 1	34.65	33.7	2.74
	Robot - 2	24.32	23.07	5.13
	Robot - 3	34.18	33	3.45
	Robot - 4	22.82	21.84	4.29
8	Robot - 1	35	33.87	3.22
	Robot - 2	25.81	25	3.13
	Robot - 3	34.19	31.98	6.46
	Robot - 4	21.74	21	3.40
9	Robot - 1	36.12	33.85	6.28
	Robot - 2	25.05	24.2	3.39
	Robot - 3	33.74	32.7	3.08
	Robot - 4	21.76	20.41	6.20
10	Robot - 1	35.89	34.2	4.70
	Robot - 2	24.87	23.33	6.19
	Robot - 3	32.9	31.99	2.76
	Robot - 4	21.6	20.21	6.43
Average time required	**Robot - 1**	**35.06**	**33.52**	**4.38**
	Robot - 2	**25.01**	**23.92**	**4.35**
	Robot - 3	**33.42**	**32.00**	**4.23**
	Robot - 4	**22.11**	**21.13**	**4.40**

From the Table 7.1-7.12, the average percentage of deviation between experimental and simulational values of both path length and navigational time for single mobile robot system using FA-PFL, FA-MGA, and FA-PFL-MGA hybrid controllers are 4.71% and 4.77%, 4.73% and 4.63%, and 4.44% and 4.38%, respectively. Similarly, the maximum obtained values for percentage of deviation in path lengths and the navigational times for multiple mobile robot system are 4.96% and 4.92%, 4.73% and 4.72%, and 4.43% and 4.40% respectively. By comparing the tabulated result, it is clear that the FA-PFL-MGA controller gives the optimal path length and navigational time.

7.8 Performance Analysis of other AI Controller with the Proposed Hybrid Controllers

In this section, the simultaneous comparison between the results of the developed hybrid controller and other navigational controller are provided to check the optimality regarding path length. One by one comparison of all three hybrid controllers is presented below. The centimetre measurement is taken as the unit on the proportional basis.Initially, the FA-PFL controller is compared with the neuro-fuzzy controller presented by Cherron et al. [202] and then it is again compared with the fuzzy-neural controller presented by the He et al. [203]. The Figure 7.20 and 7.21 show the comparison between the neuro-fuzzy controller and proposed FA-PFL hybrid controller whereas the Figure 7.22 and 7.23 compares the fuzzy-neural controllers with the proposed FA-PFL hybrid controller. The path length comparisons between these two controllers are presented in Table 7.13.

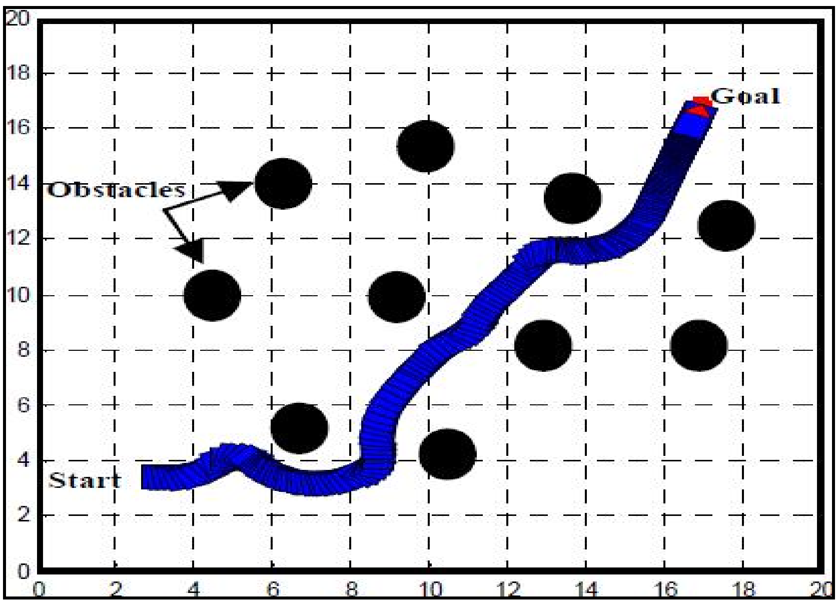


Figure 7.20: Neuro-Fuzzy Controller by Cherron et al. [202]

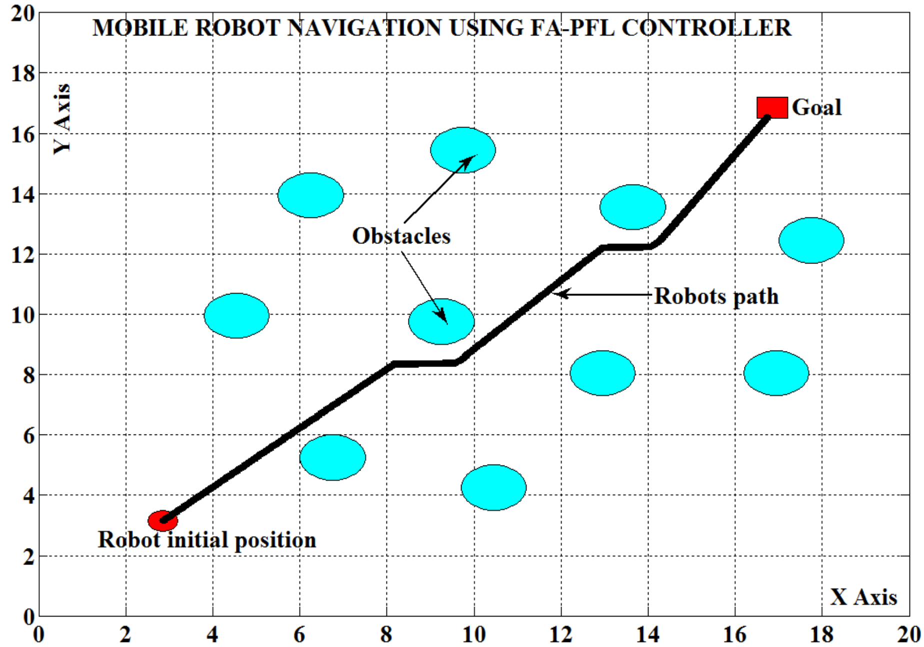


Figure 7.21: Navigation using FA-PFL hybrid controllers

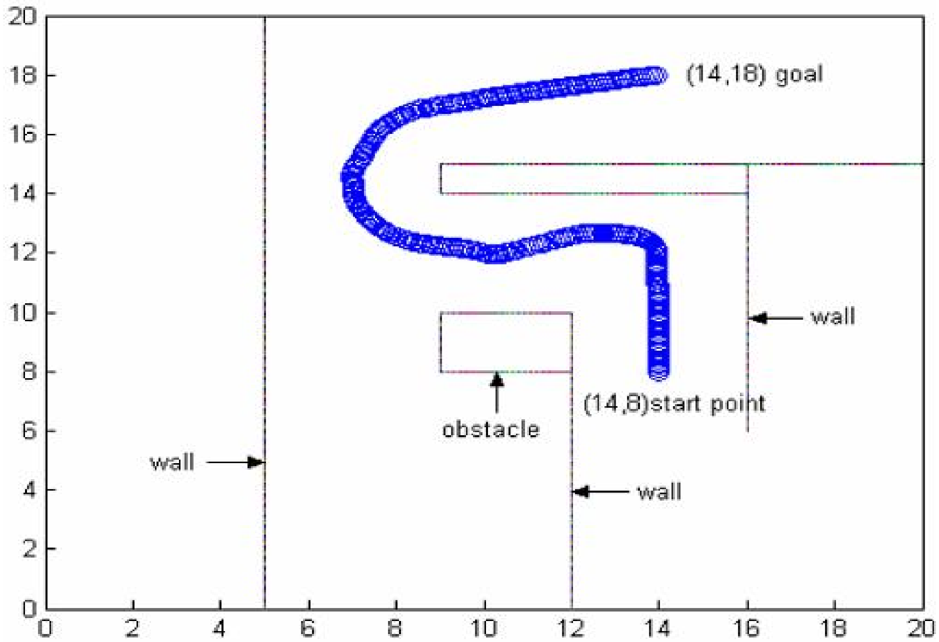


Figure 7.22: Fuzzy-Neural controllers by He et al. [203]

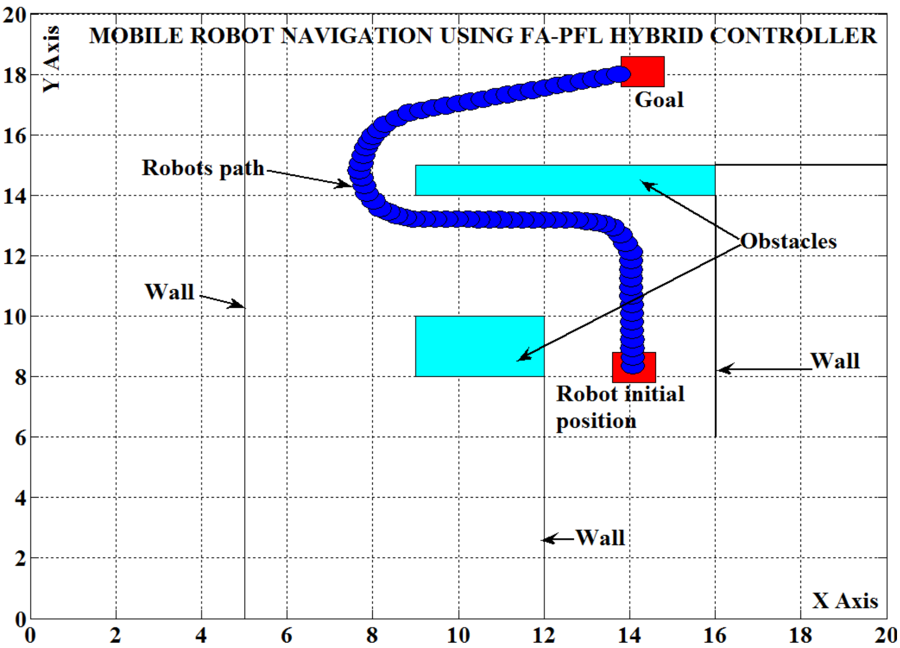


Figure 7.23: Navigation using FA-PFL hybrid controllers

Table 7.13: Comparision of simulational result regarding path length

Sl. No.	Simulational path length (in 'cm') by other AI controllers	Simulational path length (in 'cm') by FA-PFL hybrid controller	% of path length saved by FA-PFL hybrid controller
Scenario-1	10.1 (Figure 7.20)	9 (Figure 7.21)	10.89
Scenario-2	9.6 (Figure 7.22)	9 (Figure 7.23)	6.25

The comparison of the FA-MGA hybrid controller with the other AI controller such as fuzzy-neural and fuzzy logic is presented below. The Figure 7.24 shows the fuzzy-neural controller presented by the Shi et al. [204] and Figure 7.26 presented by MO et al. [205] for the MRN by using fuzzy alone. The Table 7.14 shows the path length comparison between the proposed FA-MGA hybrid controller with other AI controller such as fuzzy-neural and fuzzy logic.

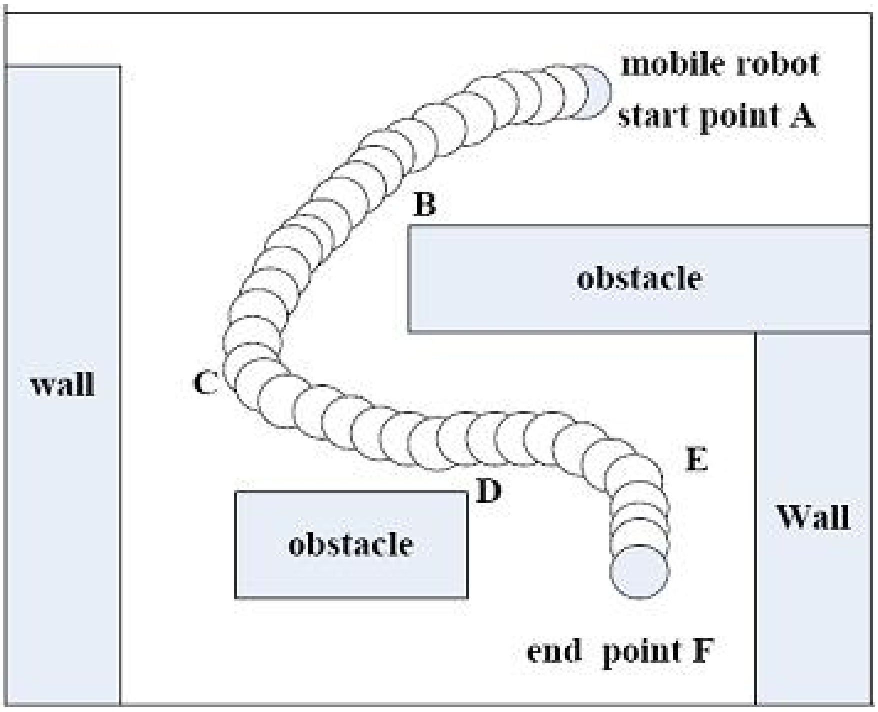


Figure 7.24: Fuzzy-Neural controller by Shi et al. [204]

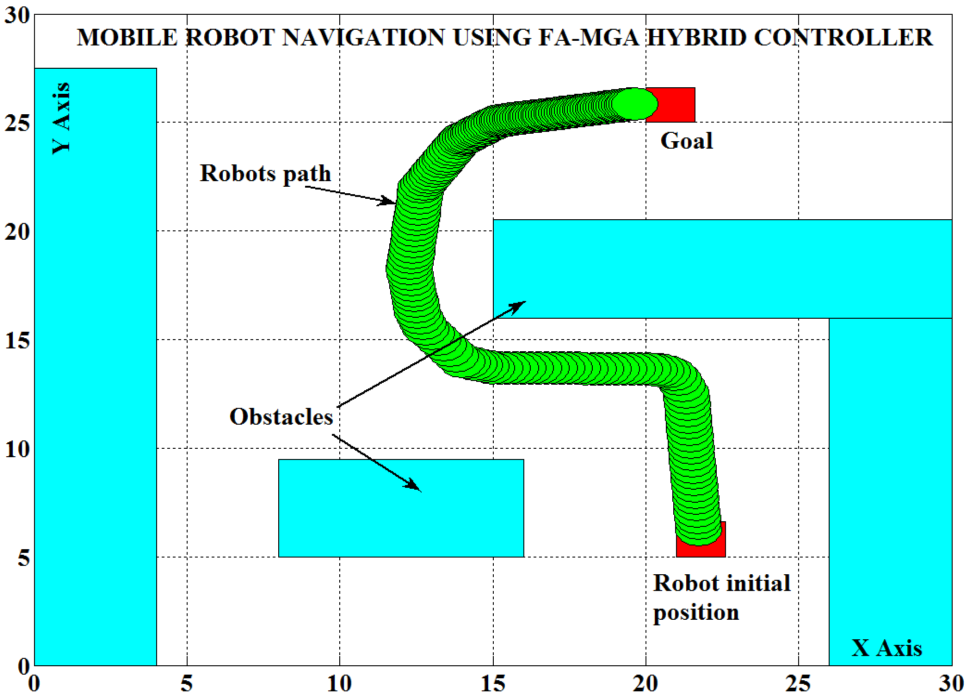


Figure 7.25: Navigation using FA-MGA hybrid controller

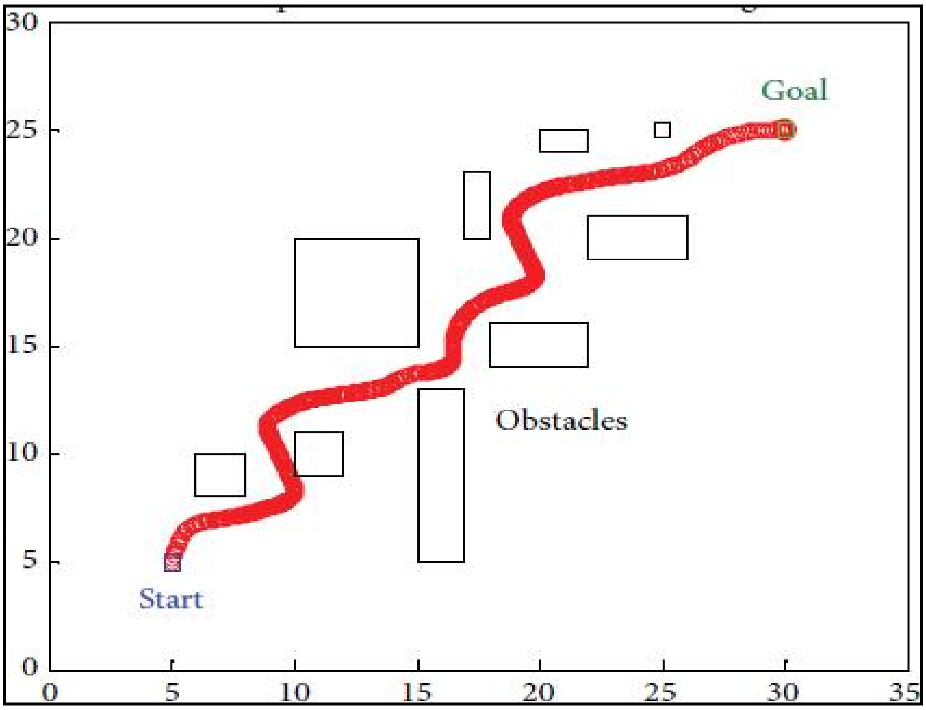


Figure 7.26: Fuzzy controller by Mo et al. [205]

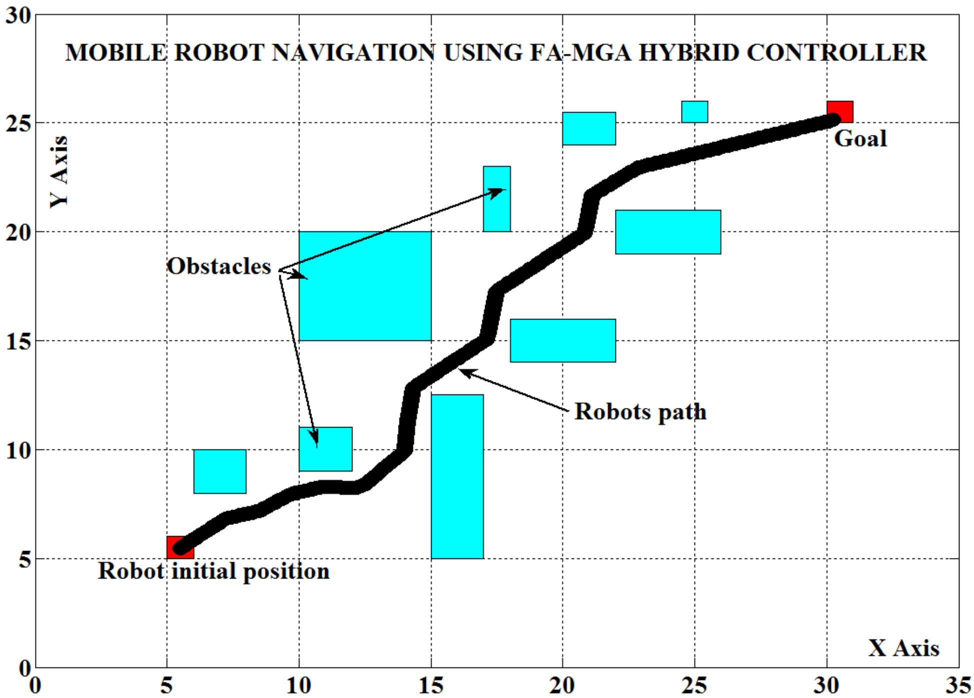


Figure 7.27: Navigation using FA-MGA hybrid controller

Table 7.14: Comparision of simulational result regarding path length

Sl. No.	Simulational path length (in 'cm') by other AI controllers	Simulational path length (in 'cm') by FA-MGA hybrid controller	% of path length saved by FA-MGA hybrid controller
Scenario-1	12 (Figure 7.24)	10.1 (Figure 7.25)	15.83
Scenario-2	10.3 (Figure 7.26)	9.3 (Figure 7.27)	9.70

At last, the comparison of the FA-PFL-MGA controller is compared with the neuro-fuzzy controller presented by Joshi et al. [206] and neural network presented by Engedy et al. [207]. The comparison of the neuro-fuzzy controller with proposed controller is shown in Figures 7.28 and 7.29 whereas the comparison of the neural network controller with the proposed controller is shown in Figures 7.30 and 7.31. The Table 7.15 gives the path length comparision in their respective environment.

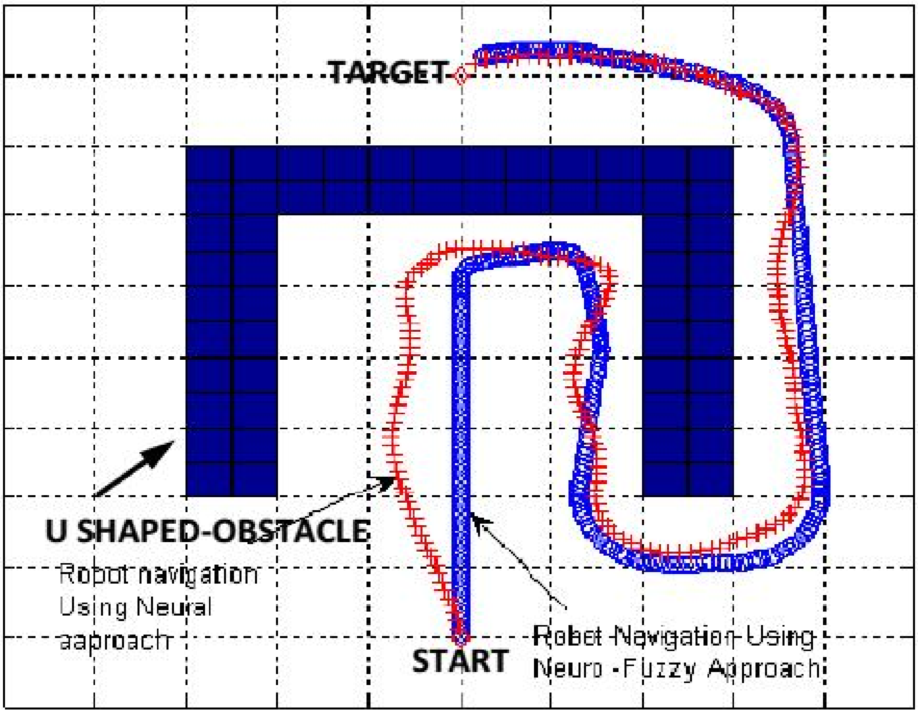


Figure 7.28: Neuro-Fuzzy Controller by Joshi et al. [206]

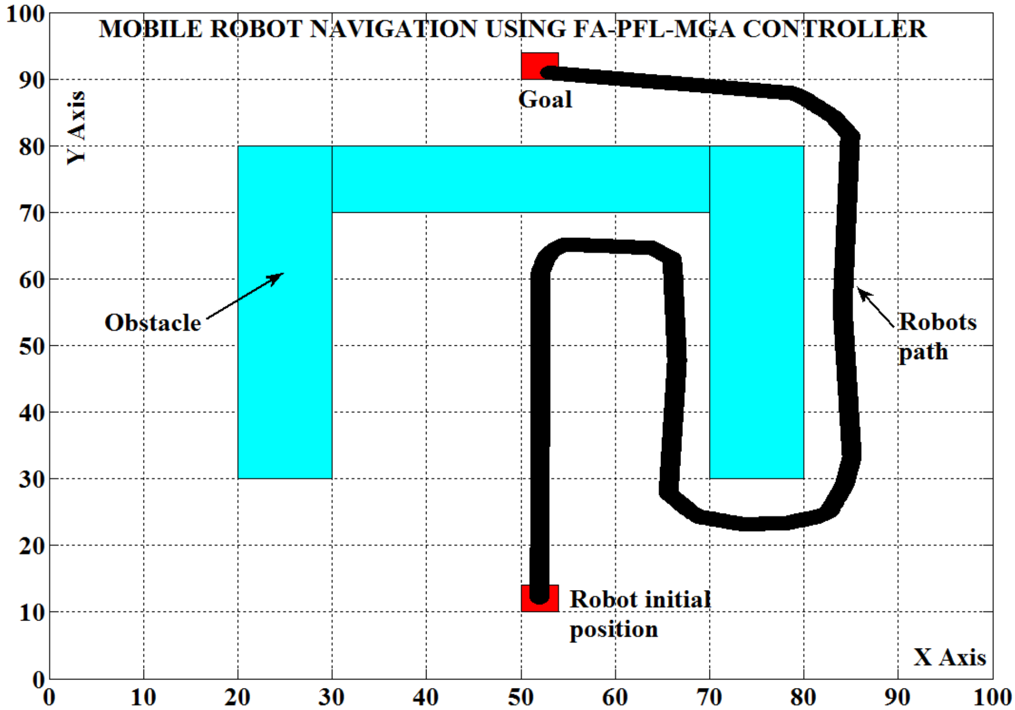


Figure 7.29: Navigation using FA-PFL-MGA hybrid controller

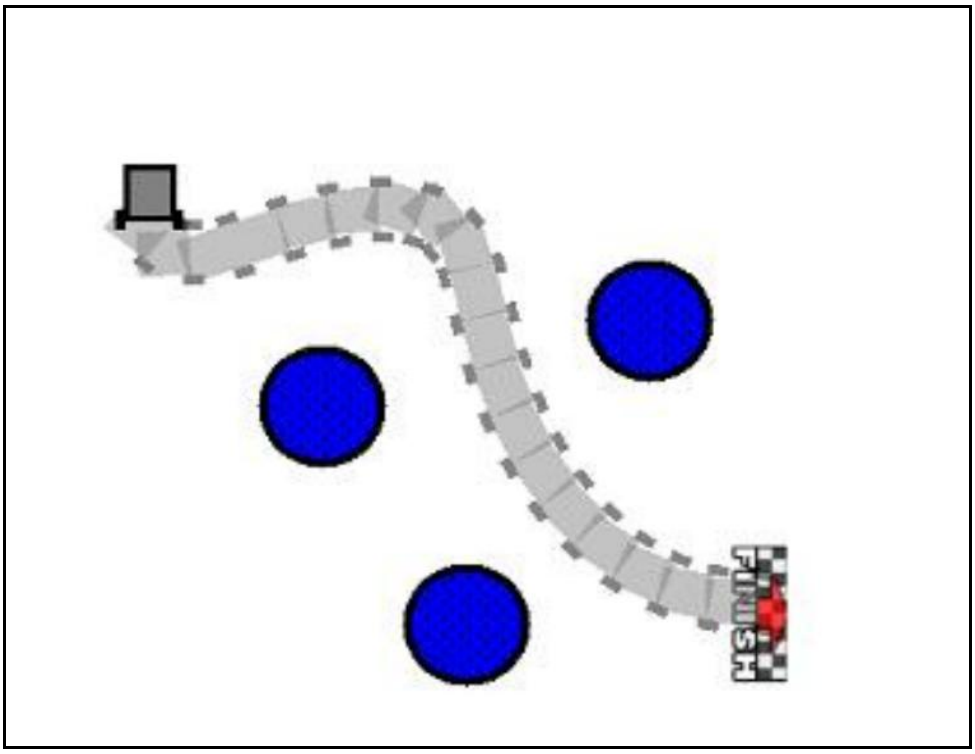


Figure 7.30: Artificial neural network controller by Engedy et al. [207]

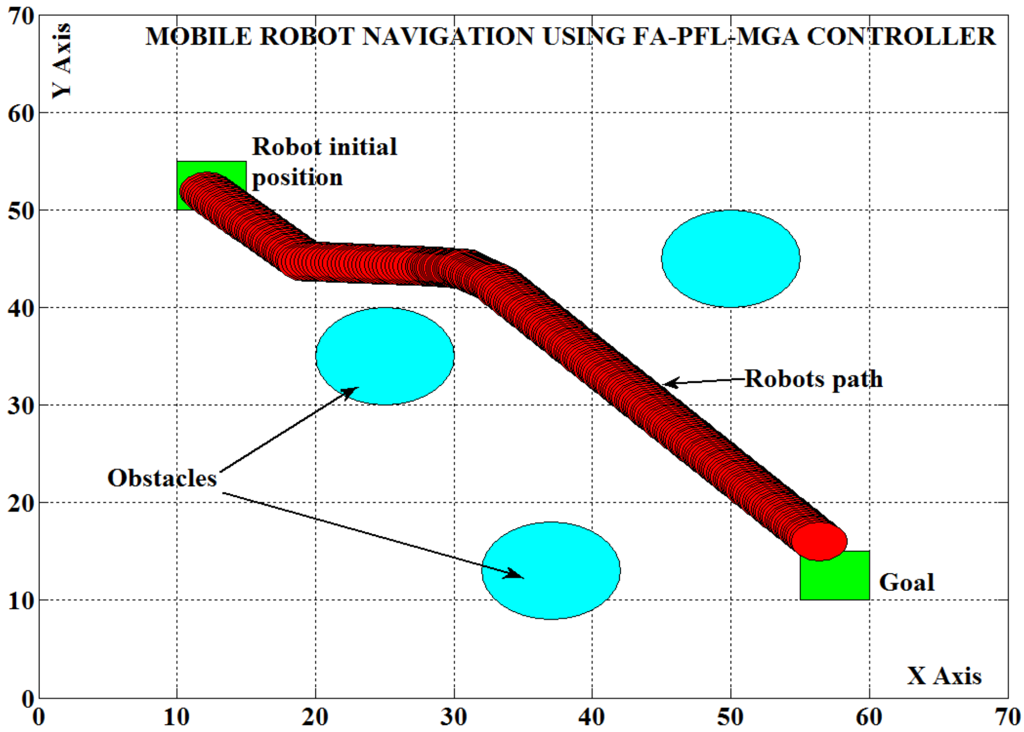


Figure 7.31: Navigation using FA-PFL-MGA hybrid controller

Table 7.15: Comparision of simulational result regarding path length

Sl. No.	Simulational path length (in 'cm') by other AI controllerS	Simulational path length by (in 'cm') FA-PFL-MGA hybrid controller	% of path length saved by FA-PFL-MGA hybrid controller
Scenario-1	21(Figure 7.28)	20.4 (Figure 7.29)	2.85
Scenario-2	9.6 (Figure 7.30)	8 (Figure 7.31)	16.66

In this section, the comparisons of the proposed hybrid controller are presented. The graphical analysis of the proposed controller with other AI controllers are presented to prove the authenticity regarding path planning. The section below analyzes the obtained results of the developed hybrid controller for wheeled mobile robot navigation.

7.9 Summary

This section presents the simultaneous study of the three developed hybrid navigational controller such as FA-PFL, FA-MGA and FA-PFL-MGA for mobile robot navigation. The above controllers perform well in various environmental conditions for the single robot and multiple robot systems. The proposed controllers are dynamic to handle the various static and dynamic environmental condition. The key features of the proposed hybrid controller are presented below:

- The proposed FA-PFL hybrid controller is successfully demonstrated for various environmental conditions in the presence of a static and dynamic obstacle. The average percentage of deviation between experimental and simulational values of both path length and navigational time for single mobile robot system using FA-PFL hybrid controller is 4.71% and 4.77%. Similarly, the maximum obtained values for percentage of deviation in path length and the navigational time for multiple mobile robot system is 4.96% and 4.92%. On comparisons with the other navigational controller such as neuro-fuzzy and fuzzy-neural, it is observed that FA-PFL saves the path length up to 10.89% and 6.25% respectively.
- The proposed FA-MGA hybrid controller is efficient to handle the problem of single and multiple mobile robot system in static and dynamic environment. From the obtained results, it has been noticed that the average percentage of deviation between experimental and simulational values of both path length and navigational time for single mobile robot system using FA-MGA hybrid controller is 4.73% and 4.63%. Similarly, the maximum obtained values for percentage of deviation in path length and the navigational time for multiple mobile robot system is 4.73% and 4.72%. By comparing the tabulated result with FA-PFL, it is clear that the FA-MGA controller gives the optimal path length and navigational time. On comparison with the other AI controller such as fuzzy-neural and fuzzy logic it saves the path length up to 15.83% and 9.70% respectively.
- The proposed FA-PFL-MGA hybrid controller is developed by combining the three different navigational approaches inorder to get optimized path length and navigational time. The three level filter of the decision parameter make it suitable to perform efficiently in known and unknown environment. The average percentage of deviation between experimental and simulational values of both path length and navigational time for single mobile robot system using FA-PFL-MGA hybrid controller is 4.44% and 4.38%. Similarly, the maximum obtained values for percentage of deviation in path length and the navigational time for multiple mobile robot system is 4.43% and 4.40%. From the tabulation, it is also clear that the percentage of deviation between simulational and experimental results for FA-PFL-MGA controller is less compared to FA-PFL and FA-MGA. Therefore, the

proposed controller can be used in the environment with maximum uncertainty. On comparisons with the neuro-fuzzy controller and neural network controller it saves path length upto 2.86% and 16.66% respectively.

- The application of the FA and LSE act as an initial filter for the all the controller, which results in path optimality and minimizes the required time of navigation.
- The obtained results prove the applicability of FA for obtaining successful hybrid technique for wheeled mobile robot navigation.

Chapter 8

Results and Discussion

8.1 Introduction

In the previous chapter, the study of various navigational controllers such as Probability Fuzzy Logic (PFL), Matrix based Genetic Algorithm (MGA), Firefly Algorithm (FA), Firefly Algorithm-Probability Fuzzy Logic (FA-PFL), Firefly Algorithm-Matrix based Genetic Algorithm (FA-PFL) and Firefly Algorithm-Probability Fuzzy Logic-Matrix based Genetic Algorithm (FA-PFL-MGA) have been analyzed for mobile robot navigation. The current chapter deals with the comparison of all discussed controller for successful mobile robot navigation. The simulational and experimental evaluations of the each controller for the path optimization are systematically presented over same environmental setup.

8.2 Investigation of Simulational and Experimental Results

In the previous chapter, we have studied various navigational controllers. These controllers are individually examined for their simulational and experimental analysis. This section presents the performance analysis of the discussed controller over same environmental conditions. The same environmental setup has been provided in case of single and multiple mobile robot navigations. The analysis is presented in both static and dynamic environment. The Figure (8.1 and 8.2) presents the simulation result by the individual controller in the static environment by using the single robot and Figure 8.3 presents simulation results for multiple wheeled mobile robot navigations. The Figure 8.4 gives the navigation of single mobile robot in the presence of the dynamic obstacle. The simulational results for optimal path length and minimum navigational time have been recorded by conducting many trials and the best path is selected for individual controller over the same environmental condition. The Table (8.1 and 8.2) clears that the hybrid controllers gives the better results regarding path optimality and navigational time over the individual controller. The hybrid controller FA-PFL-MGA performs better when compared to rest of the controller. Table 8.3 presents the comparison of the navigational controllers for the multiple mobile robot systems. The obtained result proves that the hybrid algorithm is more applicable for multiple robot systems. The performance of the robot in a dynamic environment is presented in Table 8.4 and again it has been noticed that hybrid approaches perform well in the dynamic environment.

During the simulation and experimental analysis, it is observed that the FA controller performs well when compared to the other individual controller such as PFL and MGA. The ability to handle the uncertainty and faster convergence rate make it suitable for the development of the hybrid controller and therefore the hybrid algorithm performs well over the other individual controller. The FA-PFL-MGA hybrid controller has performed well in static, dynamic and multi-robot system because of three filters of parameters being arranged in series for path length optimality and navigational time optimality.

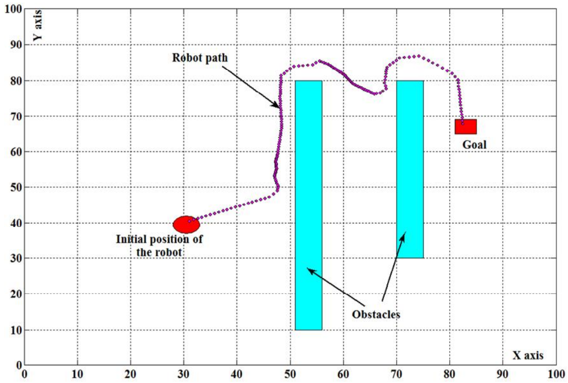


Figure 8.1 (a) Navigation using PFL

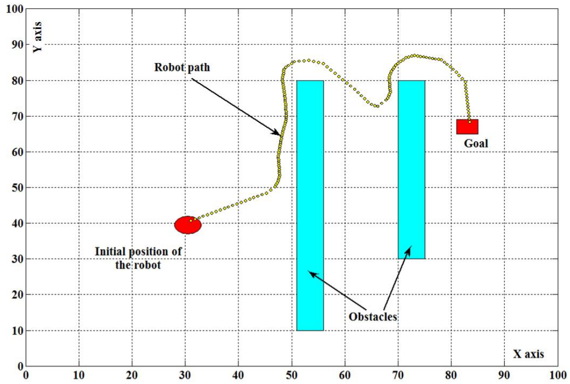


Figure 8.1 (b) Navigation using MGA

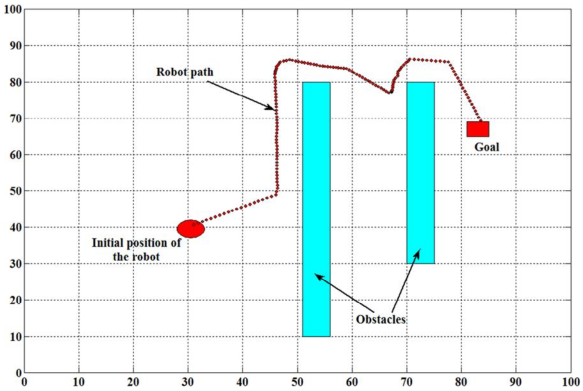


Figure 8.1(c) Navigation using FA

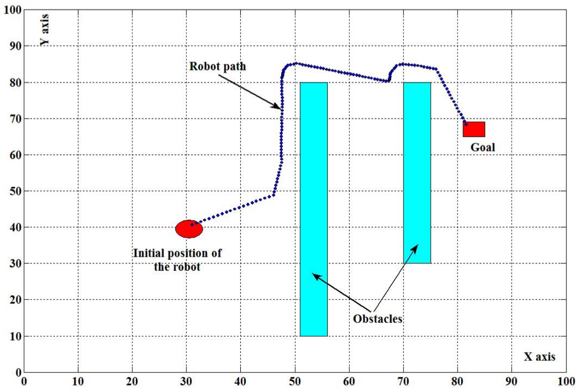


Figure 8.1(d) Navigation using FA-PFL

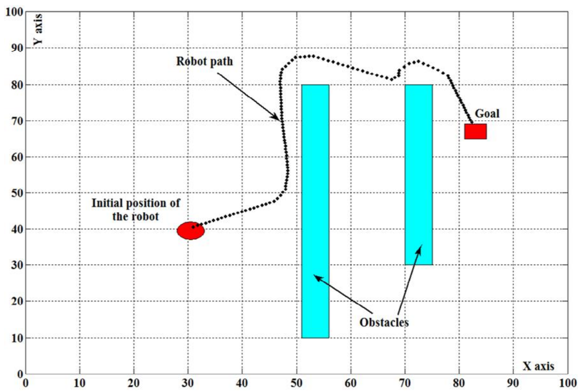


Figure 8.1(e) Navigation using FA-MGA

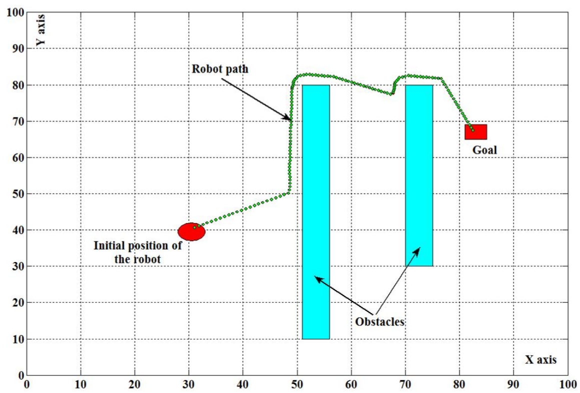


Figure 8.1(f) Navigation using FA-PFL-MGA

Figure 8.1: Navigation in static environment using single robot

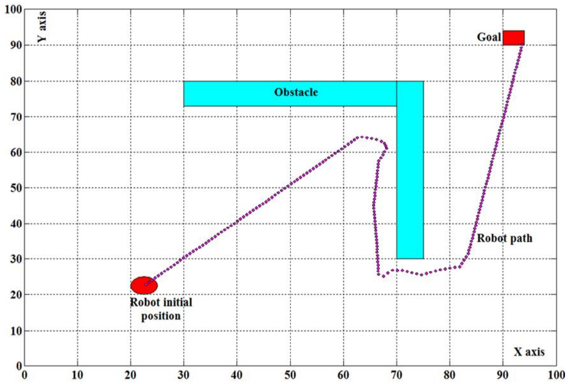


Figure 8.2(a) Navigation using PFL

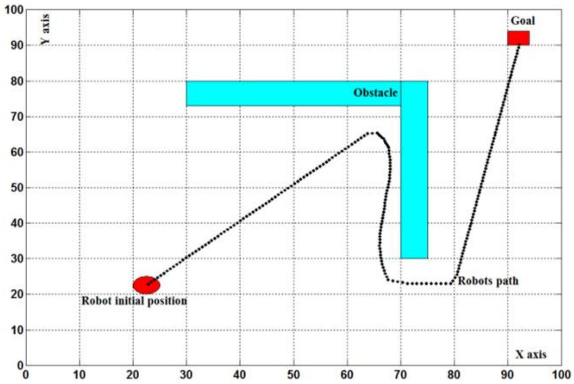


Figure 8.2(b) Navigation using MGA

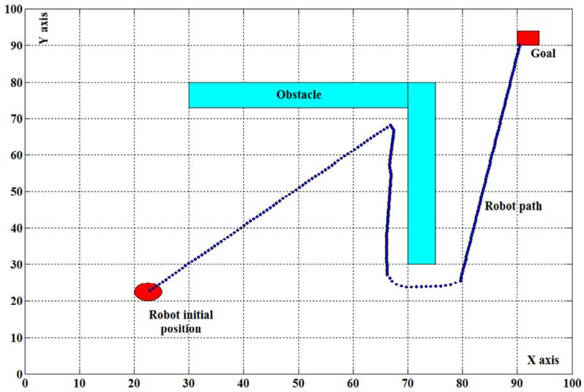


Figure 8.2(c) Navigation using FA

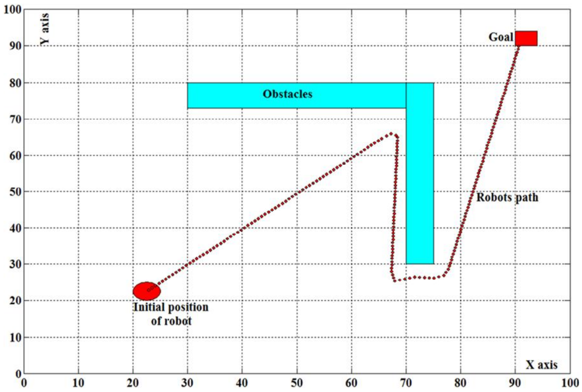


Figure 8.2(d) Navigation using FA-PFL

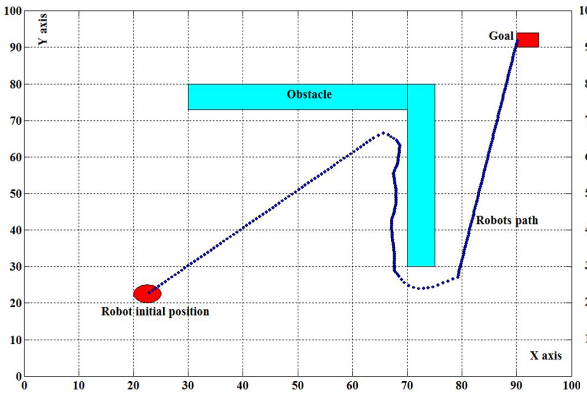


Figure 8.2(e) Navigation using FA-MGA

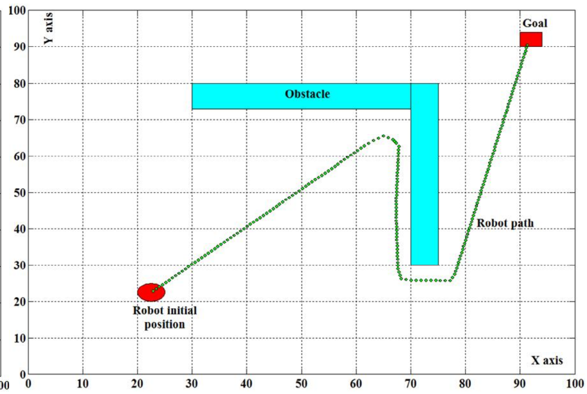


Figure 8.2(f) Navigation using FA-PFL-MGA

Figure 8.2: Navigation in static environment using single robot

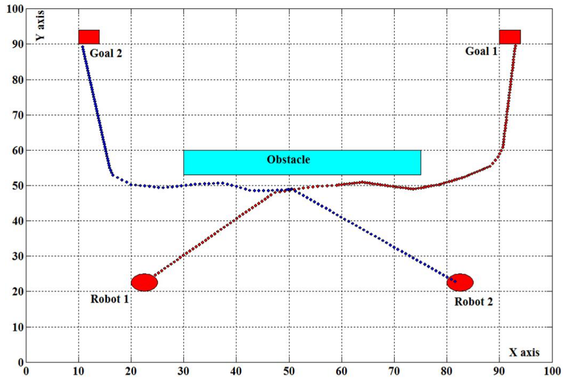


Figure 8.3(a) Navigation using PFL

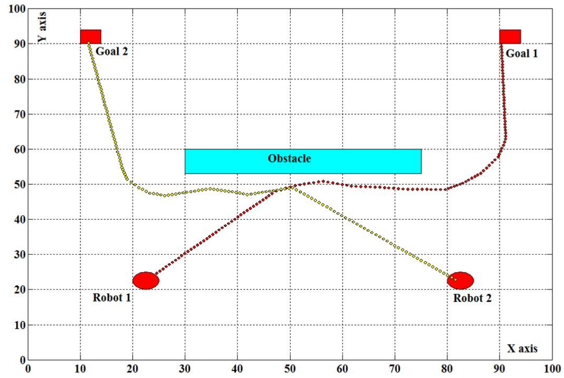


Figure 8.3(b) Navigation using MGA

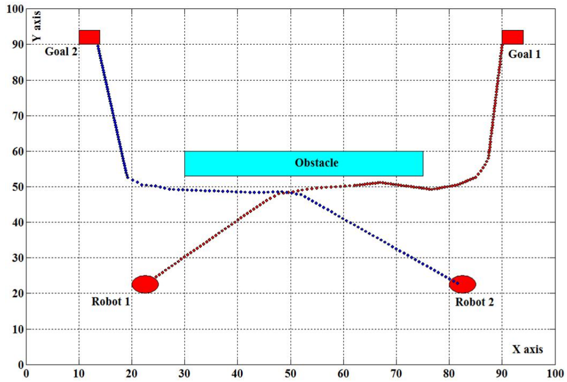


Figure 8.3(c) Navigation using FA

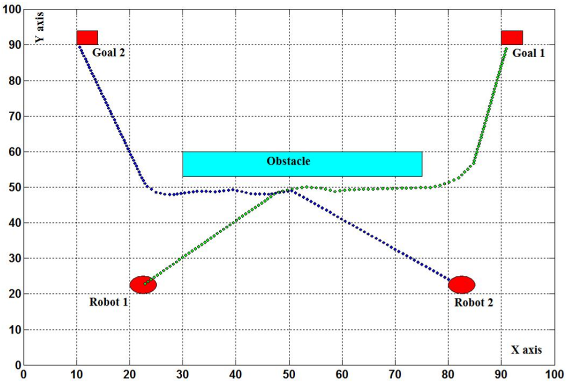


Figure 8.3(d) Navigation using FA-PFL

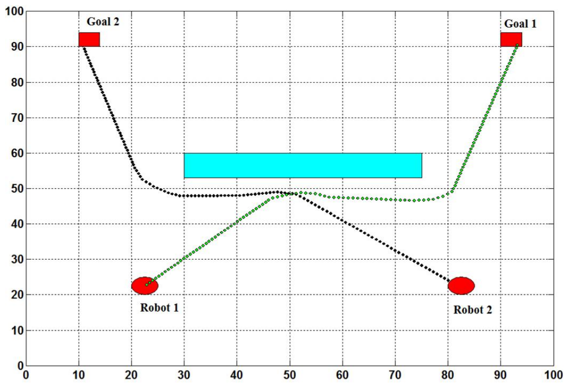


Figure 8.3(e) Navigation using FA-MGA

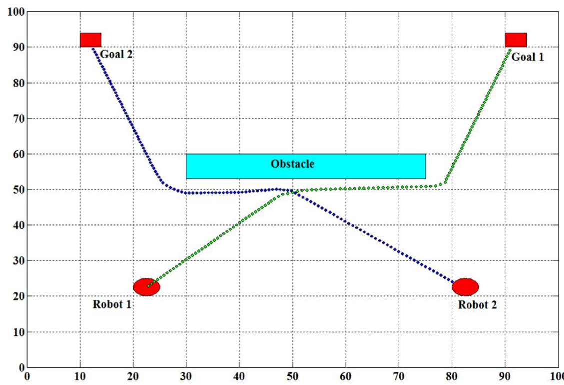


Figure 8.3(f) Navigation using FA-PFL-MGA

Figure 8.3: Navigation in static environment using multiple robots

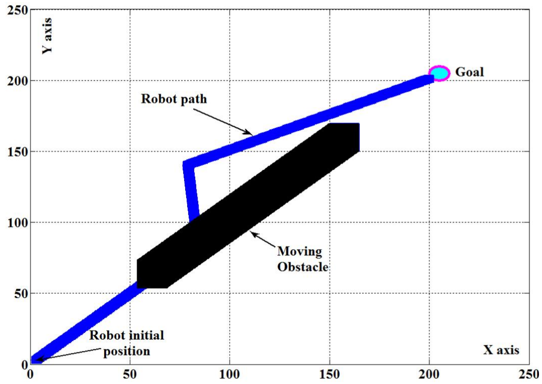


Figure 8.4(a) Navigation using PFL

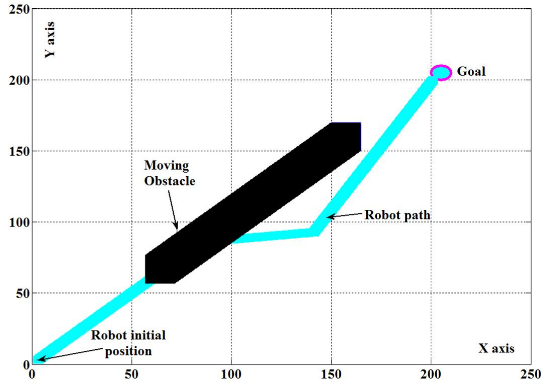


Figure 8.4(b) Navigation using MGA

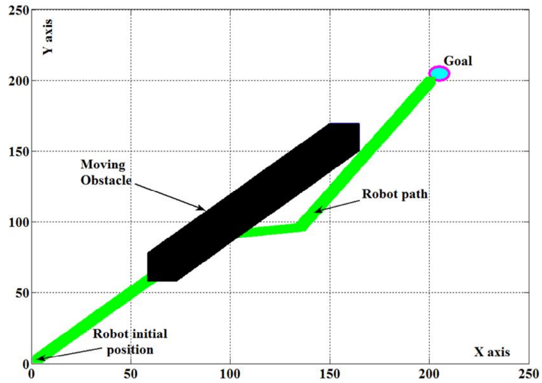


Figure 8.4(c) Navigation using FA

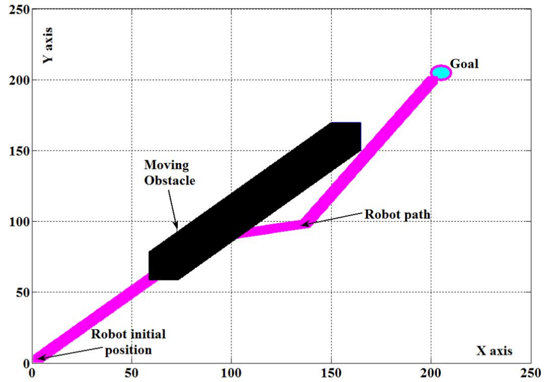


Figure 8.4(d) Navigation using FA-PFL

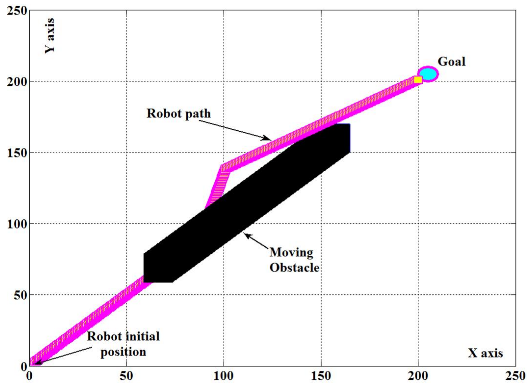


Figure 8.4(e) Navigation using FA-MGA

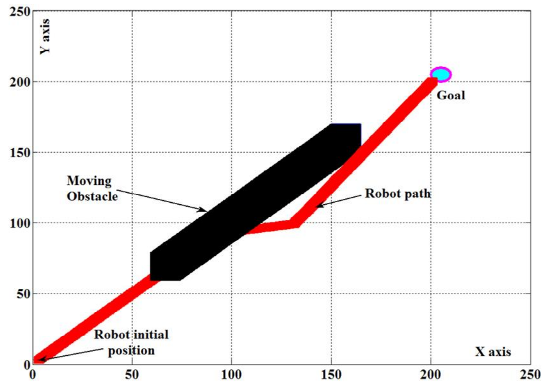


Figure 8.4(f) Navigation using FA-PFL-MGA

Figure 8.4: Navigation in dynamic environment

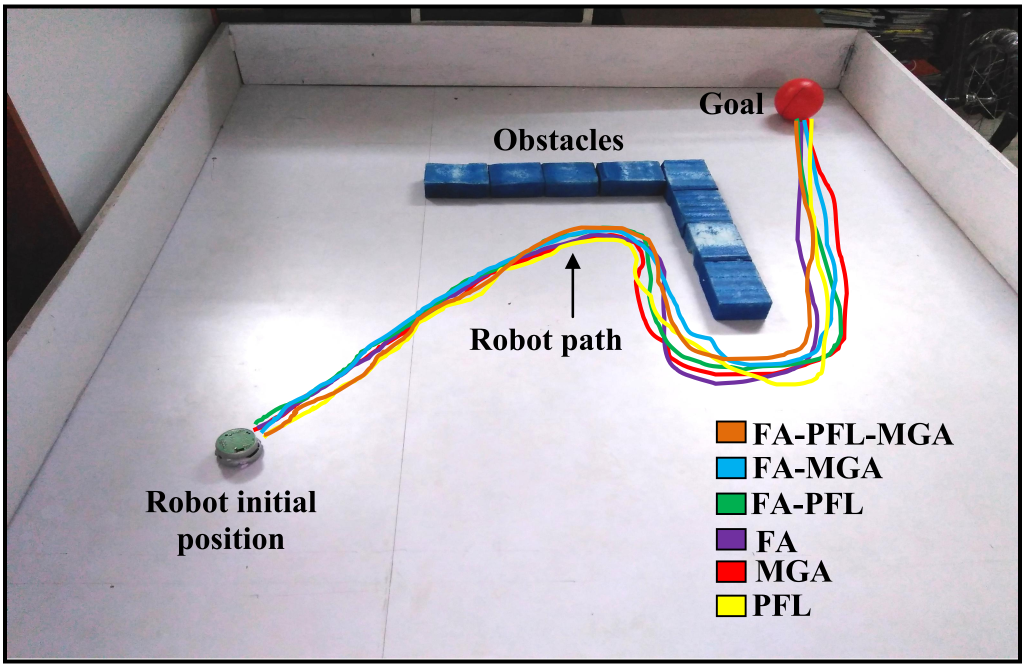


Figure 8.5: Real-time navigation of mobile robot using developed controllers

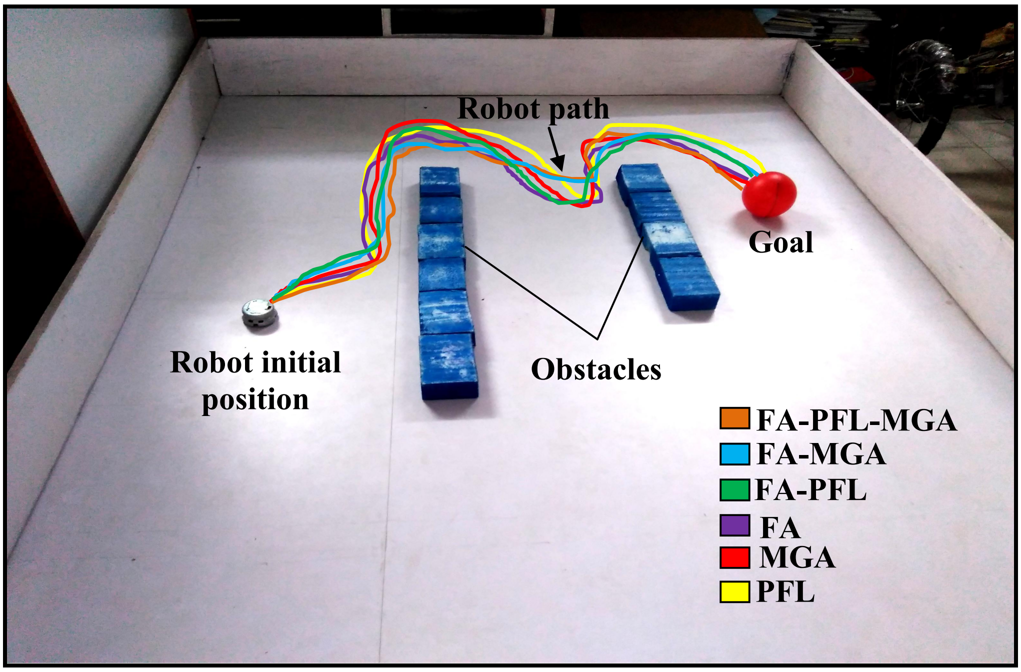


Figure 8.6: Real-time navigation of mobile robot using developed controllers

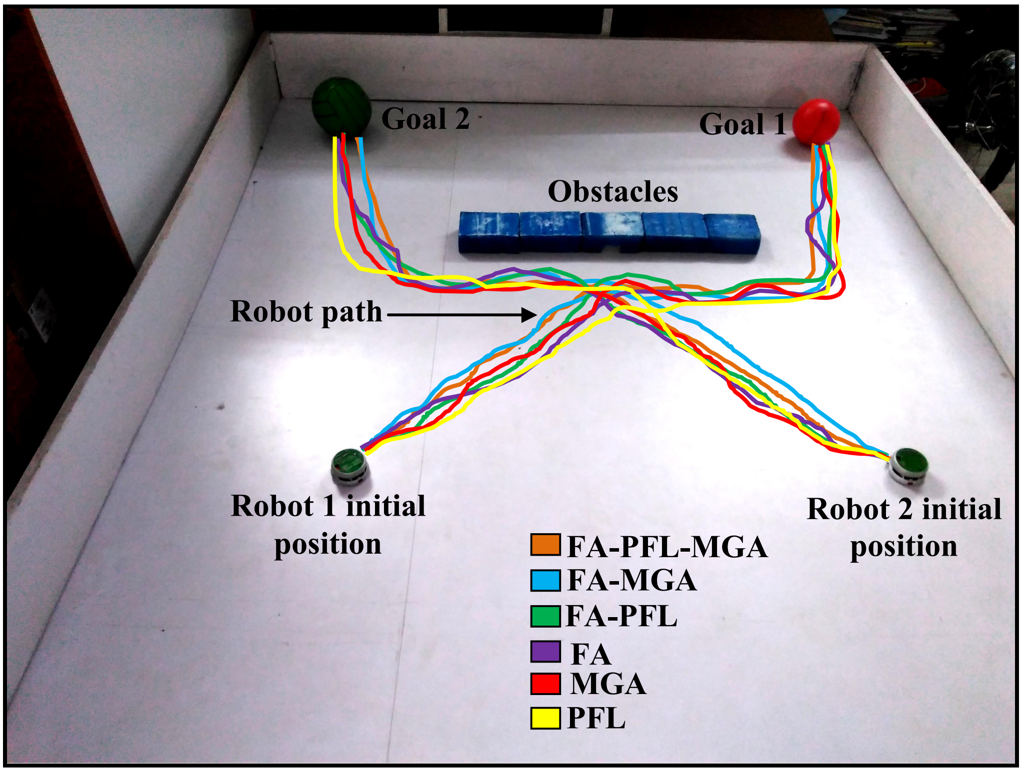


Figure 8.7: Real-time navigation of multiple mobile robots using developed controllers

Table 8.1: Path length comparison over similar environmental setup (Figure 8.1 and 8.5)

Sl. No.	Name of controllers	Experimental path length (in 'cm')	Simulational path length (in 'cm')	% of deviation
1	PFL	170.1	161.09	5.29
2	MGA	164.7	156.17	5.17
3	**FA**	**159.3**	**151.38**	**4.97**
4	FA-PFL	153.9	146.57	4.76
5	FA-MGA	145.8	138.95	4.69
6	**FA-PFL-MGA**	**143.1**	**136.68**	**4.48**

Table 8.2: Navigational time comparison over similar environmental setup (Figure 8.1 and 8.5)

Sl. No.	Name of controllers	Experimental time (in 'sec')	Simulational time (in 'sec')	% of deviation
1	PFL	27.42	25.96	5.32
2	MGA	26.54	25.17	5.16
3	**FA**	**25.67**	**24.4**	**4.94**
4	FA-PFL	24.8	23.62	4.75
5	FA-MGA	23.5	22.39	4.72
6	**FA-PFL-MGA**	**23.06**	**22.03**	**4.46**

Table 8.3: Path length comparison over similar environmental setup (Figure 8.2 and 8.6)

Sl. No	Name of controllers	Experimental path length (in 'cm')	Simulational path length (in 'cm')	% of deviation
1	PFL	294.3	279	5.19
2	MGA	286.2	271.6	5.10
3	**FA**	**280.8**	**267.2**	**4.87**
4	FA-PFL	275.4	262.5	4.68
5	FA-MGA	264.6	252.3	4.64
6	**FA-PFL-MGA**	**251.1**	**239.7**	**4.54**

Table 8.4: Navigational time comparison over similar environmental setup (Figure 8.2 and 8.6)

Sl. No.	Name of controllers	Experimental Time (in 'sec')	Simulational Time (in 'sec')	% of deviation
1	PFL	47.44	44.97	5.20
2	MGA	46.13	43.78	5.09
3	**FA**	**45.28**	**43.07**	**4.88**
4	FA-PFL	44.39	42.31	4.68
5	FA-MGA	42.65	40.67	4.64
6	**FA-PFL-MGA**	**40.47**	**38.63**	**4.54**

Table 8.5: Path length comparison over similar environmental setup (Figure 8.3 and 8.7)

Sl. No.	Name of controllers		Experimental path length (in 'cm')	Simulational path length (in 'cm')	% of deviation
1	PFL	Robot -1	180.9	171.5	5.19
		Robot-2	197.1	186.7	5.27
2	MGA	Robot -1	175.5	166.59	5.07
		Robot-2	193.05	183.15	5.12
3	**FA**	Robot -1	**172.8**	**164.21**	**4.97**
		Robot-2	**189**	**179.47**	**5.04**
4	FA-PFL	Robot -1	167.4	159.23	4.88
		Robot-2	189	179.61	4.96
5	FA-MGA	Robot -1	164.7	156.82	4.78
		Robot-2	186.3	177.13	4.92
6	**FA-PFL-MGA**	Robot -1	**162**	**154.47**	**4.64**
		Robot-2	**183.6**	**174.68**	**4.85**

Table 8.6: Navigational time comparison over similar environmental setup (Figure 8.3 and 8.7)

Sl. No.	Name of controllers		Experimental time (in 'sec')	Simulational time (in 'sec')	% of deviation
1	PFL	Robot -1	29.12	27.64	5.08
		Robot-2	31.77	30.06	5.38
2	MGA	Robot -1	28.29	26.85	5.09
		Robot-2	31.11	29.52	5.11
3	**FA**	Robot -1	**27.85**	**26.47**	**4.95**
		Robot-2	**30.46**	**28.93**	**5.02**
4	FA-PFL	Robot -1	26.98	25.66	4.89
		Robot-2	30.46	28.95	4.95
5	FA-MGA	Robot -1	26.54	25.27	4.78
		Robot-2	30.03	28.55	4.92
6	**FA-PFL-MGA**	Robot -1	**26.11**	**24.9**	**4.63**
		Robot-2	**29.59**	**28.15**	**4.86**

Table 8.7: Path length comparison in dynamic environment Figure 8.4

Sl. No.	Name of controllers	Simulational path length (in 'cm')	Simulational time (in 'sec')
1	PFL	205.21	33.07
2	MGA	197.15	31.78
3	**FA**	**194.46**	**31.34**
4	FA-PFL	189.44	30.53
5	FA-MGA	180.9	29.12
6	**FA-PFL-MGA**	**170.17**	**27.43**

8.3 Summary

The current chapter introduces the PFL, MGA, FA, FA-PFL, FA-MGA and FA-PFL-MGA controllers for mobile robot navigation. The performance analysis of proposed controller regarding path length and navigational time is presented here for various environmental conditions. The important findings are given as:

- The proposed controller is good enough to avoid the obstacles in the static and dynamic environment.
- The proposed controller successfully handles the problem of multiple wheeled mobile robot navigations.
- The proposed controller avoids the random moving of the robot in the environment and assures safe and near optimal path planning.
- The comparative study of the individual controllers such as PFL, MGA and FA state that the FA controller gives the better agreement over path optimality and minimizes the navigational time from start to goal position of the robot.
- The application of the FA for hybridization with other controller provides the better results over the FA controller. The hybrid controller such as FA-PFL, FA-MGA and FA-PFL-MGA are more effective compared to individual FA controller.
- The three level FA-PFL-MGA controller is found most effective among all the controllers.
- By using the hybrid controller we can keep the percentage of deviation between the experimental and simulation results within the 5%.

At last, from the obtained results we can say that the development of the hybrid algorithm is much effective for mobile robot navigation than a standalone controller.

Chapter 9

Conclusions and Future Directions

Previous chapters give the study of navigation and the control strategies for wheeled mobile robots by using various navigational controllers in uncertain environments. This chapter sums up the key contributions of the proposed research work and also gives the further investigation which is to be carried in future.

9.1 Contribution of the Proposed Work

The primary goal of the research work is a concern with the design and implementation of the artificial intelligent controller for solving the navigational path planning problem of multiple wheeled mobile robot systems. The artificial intelligent controller is developed to handle the uncertainty present in the environment. The work presented in the thesis highlights the autonomous navigation of the mobile robot without having prior knowledge of the static and dynamic obstacles. The proposed navigational strategies for autonomous mobile robot navigation are Firefly algorithm (FA), Probability Fuzzy Logic (PFL) and Matrix based Genetic Algorithm (MGA). The hybrid approaches such as FA-PFL, FA-MGA and FA-PFL-MGA are also presented to enhance the performance of the standalone controller for navigational issues. The main contributions of the research work in the field of wheeled mobile robot navigation are presented below:

- The study of the kinematics of the Wheeled Mobile Robot is presented to understand the factor which influences the navigation of the robot and how the velocity analysis of individual wheel is needed to get required heading angle for navigation.
- The application of the probability for fuzzy logic is presented for selection of the best decision rule for getting required steering angle of navigation.
- The application of the matrix- trace based arrangement boost the performance of the genetic algorithm. It transforms the GA into small sample space from the large sample space.

- The Firefly Algorithm is introduced for optimal path planning of the wheeled mobile robot in the static and dynamic environments.
- The FA based new hybrid controllers such as FA-PFL, FA-MGA and FA-PFL-MGA are presented for the path planning strategies of multiple mobile robot systems.
- The developed controller is initially tested on the Matlab software for simulation analysis and then it is implemented for the real-time analysis over the Khepera-II robots. The observed simulational and experimental result for proposed controller show the good agreement regarding optimal path planning.

9.2 Conclusions

The results obtained from a series of simulation and experimental investigation; it is found that all the proposed controllers can solve the issues regarding the navigation of mobile robots. The proposed controllers have been tested for single and multiple wheeled robots with same and different goal positions in various environments.

- The Tables 8.1-8.7 shows that the nature inspired metaheuristic firefly algorithm performs better in terms of navigational path length and navigational time when compared to standalone controllers like PFL and MGA over same environmental setup. From the Figure 8.1 and 8.5 and Table 8.1 and 8.2, it is clear that the FA saves the simulational path length by 3.27% and 6.34% when compared to MGA and PFL respectively. Similarly, in experimental observation, the path length is saved using FA is 3.16% and 6.14% when compared to MGA and PFL respectively. The navigational time required to accomplish the task in simulational and experimental environment by using the FA is less when compared to MGA and PFL. The navigational time saved by FA in simulational environment is 3% and 6% when compared to MGA and PFL respectively. Similarly, in experimental observation, the navigational path is saved using FA is 3.2% and 6.38% when compared to MGA and PFL. In the Figure 8.3 and 8.7, the percentage of deviation in path length using FA is 4.97 % for robot-1 and 5.04% for robot-2 which is comparatively less than the percentage of deviation observed in the MGA and PFL. Similarly, the percentage of deviation in navigational time for FA is 4.95% for robot-1 and 5.02% for robot-2 which is also comparatively less than the MGA and PFL. The performance of the navigational controller in dynamic environment is shown in Figure 8.4 and from the Table 8.7 it is clear that the path length required to accomplish the task by FA is 194.46 cm whereas the path length required accomplishing the same task by MGA and PFL are 197.15 cm and 205.21 cm respectively. The time required for navigation in dynamic environment by FA is 31.34 Seconds which is comparatively less than the MGA and PFL controllers i.e. 31.78 sec and 33.07 sec. respectively.

- The simulational and experimental results shown in Figure 8.1-8.7 state that the path obtained by the hybrid controllers such as FA-PFL, FA-MGA and FA-PFL-MGA is optimal than the individual controller (FA,PFL and MGA) for single and multiple mobile robot system in static and dynamic environment. According to Table 8.1-8.7, the data reflects that the hybrid controller is able to generate the shortest path in minimum time.
- From the Table 7.1-7.12, the average percentage of deviation between experimental and simulational values of both path length and navigational time for single mobile robot system using FA-PFL, FA-MGA, and FA-PFL-MGA hybrid controllers are 4.71% and 4.77%, 4.73% and 4.63%, and 4.44% and 4.38%, respectively. Similarly, the maximum obtained values for percentage of deviation in path lengths and the navigational times for multiple mobile robot system are 4.96% and 4.92%, 4.73% and 4.72%, and 4.43% and 4.40% respectively. By comparing the tabulated result, it is clear that the FA-PFL-MGA controller gives the optimal path length and navigational time. The Figure 8.1-8.7 show that the obtained path by the FA-PFL-MGA is small and data reflects in Table 8.1-8.7 confirms that the path length and navigational time is minimum compared to PFL, MGA, FA, FA-PFL, FA-MGA controllers.
- The percentage of deviations for simulational and experimental results for PFL, MGA, FA, FA-PFL, FA-MGA and FA-PFL-MGA controllers are less than 7%. This shows that the proposed controllers performs better.

9.3 Future Directions

The section presents the future scope of the present work as follows:

- In the proposed work, the developed controllers is tested in indoor environment however this controller can be tested in an outdoor environment.

- The developed controllers can be implemented with other algorithm to enhance the performance of the wheeled mobile robot.
- The proposed work can be extended to the design and develop a controller to solve the dynamic goal problem.
- The navigation of the mobile robot is demonstrated on the plane ground, in future it may be demonstrated in the underwater and aerial conditions.
- The multiple mobile robots navigations in a dynamic environment can be analyzed for the autonomous driverless vehicle.

Appendix A

A1: Specification of Khepera-II robot.

Khepera-II robot used in the experiment.

A1: Specification of the Khepera-II robot used in the experiment

Sl. No.	Elements	Technical specification
1	Processor	Motorola 68331 CPU, 25MHz
2	RAM	512KB
3	Flash	512 KB
4	Motors	2-DC brushed Servo motors with incremental encoders
5	Sensors	8 Infrared proximity and ambient light sensors with up to 100mm range.
6	Speed	Max: 0.5m/s, Min:0.02m/s
7	Power	Power adapter or Rechargeable NiMH Batteries
8	Communication	Standard Serial Port, up to 115KB/S
9	Size	Diameter: 70 mm , Height: 30 mm
10	Weight	Approx. 80 g
11	Payload	Approx. 250g
12	Remote control Software via tether or radio	LabVIEW ® (on PC, MAC or SUN) using RS232 MATLAB ® (on PC, MAC, Linux or SUN) using RS232 SysQuake ® (on PC, MAC, Linux or SUN) using RS232 Freeware Any other software capable of RS232 communication

Bibliography

1. S. A. Stoeter and N. Papanikolopoulus, “Kinematic motion model for jumping scout robots,” *IEEE Transcation on Robotics*, vol. 22, no. 2, pp. 398-403, 2006.
2. K. Hosada, T. Tauma, A. Nakamoto and S. Hayashi, “Biped Robot design powered by antagonistic pneumatic actuators for multi-modal locomotion,” *Robotics and Autonomous Systems*, vol. 56, no. 1, pp. 46-53, 2008.
3. X. Wu and S. Ma, “CPG-based control of serpentine locomotion of a snake-like robot,” *Mechatronics*, vol. 20, no. 2, pp. 326-334, 2010.
4. Z. Y. Baryaktaroghu, “Snake like locomotion: Experimental with a biologically inspired wheel-less snake robot,” *Mechanism and Machine Theory*, vol. 44, no. 3, pp. 591-602, 2009.
5. J. L. Guzman, M. Berenguel, F. Rodriguez and S. Dormido, “An interactive tool for mobile robot motion planning,” *Robotics and Autonomous Systems*, vol. 56, no. 5, pp. 396-409, 2008.
6. N. Chakraborty and A. Ghoshal, “Kinematics of wheeled mobile robots on uneven terrain,” *Mechanism and Machine Theory*, vol. 39, no. 12, pp. 1273-1287, 2004.
7. G. Campion, G. Bastin and B. D. AndraCa-Novel, “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47-62, 1996.
8. D. S. Kim, W. H. Kwon and H. S. Park, “Geometric kinematics and application of a mobile robot, International Journal of control,” *Automation and Systems*, vol. 1, no. 3, pp. 376-384, 2003.
9. J. H. Lee, B. K. Kim, T. Tanikawa and K. Ohba, “Kinematics analysis on omnidirectional mobile robot with double-wheel-type active casters,” *IEEE International Conference on Control, Automation and Systems,Seoul, Korea*, vol. 17, no. 20, pp. 1217-1221, Oct 2007.
10. G. Indiveri, J. Paulus and P. G. Ploger, “Motion control of Swedish wheeled mobile robots in the presence of actuator saturation,” *LNCS 4434, Springer Berlin Heidelberg*, pp. 35-46, 2007.
11. K. Tadakuma and R. Tadakuma, “Mechanical design of “ Omni-Ball”: spherical wheel for holonomic omnidirectional motion,” *Third Annual IEEE Conference on Automation Science and Engineering, Scottsdale, USA*, vol. 22, no. 25, pp. 788-794, Sep 2007.
12. C. Cariou, R. Lenain, B. Thuilot and P. Martinet, “Adaptive control of four-wheel-steering off-road mobile robots: Application to path tracking and heading control in presence of sliding,” *IEEE International Conference on Intelligent Robots and Systems, France*, vol. 22, no. 26, pp. 1759-1764, Sep 2008.
13. F. R. Marcovitz and A. Kelly, “On line mobile robot model identification using integrated perturbative dynamics,” *Experimental Robotics*, *Springer Berlin Heidelberg*, pp. 417-431, Jan 2014.
14. J. C. Alexander and J. H. Maddocks, “On the kinematics of wheeled mobile robots,” *International Journal of Robotic Research,* vol. 8, no. 5, pp. 15-27, 1989.

15. K. Tsuchiya, Urakubo and K. Tsujita, "A motion control of a two-willed mobile robot, Systems, Man and Cybernetics," *IEEE International Conference, Tokyo*, vol. 5, pp. 690-696, 1999.
16. M. Gyula, "Introduction to control of mobile robots," *Proceedings of the YUINFO KOpaonik Serbia*, pp. 1-4, *2006.*
17. D. S. Kim, W. H. Kwon, H. S. Park, "Geometric Kinematics and Applications of a Mobile Robot," *International Journal of Control, Automation, and Systems*, vol. 3, no. 1, Sep 2003.
18. G. Indiveri, "Swedish Wheeled Omnidirectional Mobile Robots: Kinematics Analysis and Control," *IEEE Transactions on robotics*, vol. 25, no. 1, Feb 2009.
19. M. Wada, A. Takagi, S. Mori, "Caster Drive Mechanisms for Holonomic and Omnidirectional Mobile Platforms with no over Constraint," *In the Proceedings of the 2000 IEEE International Conference on Robotics and Automation San Francisco*, pp. 1531-1538, Apr 2000.
20. W. Kim, D. H. Kim, B. j. Yi, B. J. You, "Kinematic Modeling of Mobile Robots by Transfer Method of Augmented Generalized Coordinates," *Proceedings of the 2001 IEEE International Conference on Robotics and Automation Seoul, Korea*, vol. 21, no. 26, pp. 2413-2418, May 2001.
21. D. H. Shin, K. H. Park, "Velocity Kinematic Modeling for Wheeled Mobile Robots," *Proceedings of the 2001 IEEE International Conference on Robotics and Automation Seoul, Korea.* vol. 21, no. 26, pp. 3516-3522, May 2001.
22. P. F. Muir, C. P. Neuman, "Kinematic Modeling of Wheeled Mobile Robot," *Journal of Robotic System*, vol. 4, no. 2, pp. 281-340, April 1987.
23. J. Borenstein, "Control and Kinematic Design of Multi-Degree-Of-Freedom Mobile Robots with Compliant Linkage," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 21-35, Feb 1995.
24. N. Chakraborty, A. Ghosal, "Kinematics of wheeled mobile robots on uneven terrain," *Mechanism and Machine Theory*, vol. 39, pp. 1273-1287, 2004.
25. R. Rashid, I. Elamvazuthi, M. Begam and M. Arrofiq, "Fuzzy-based navigation and control of nonholonomic mobile robot," *Journal of Computing*, vol. 2, no. 3, pp. 130-137, 2010.
26. F. L. Menn, P. Bidaud and F. B. Amar, "Genetic differential kinematic modeling of articulated multi-monocycle mobile robots," *IEEE International Conference on Robotics and Automation Systems, Orlando, Florida*, vol. 15, no. 19, pp. 1505-1510, May 2006.
27. H. Teimoori and V. S. Andrey, "Equiangular and guidance of a wheeled mobile robot based on range-only measurement," *Robotics and Autonomous systems*, vol. 58, pp. 203-205, 2010.
28. Z. Cao, Y. Zhao and Y. Fu, "Research on point stabilization of a wheeled mobile robot using fuzzy control optimized by GA," *The Journal of China University of Posts and Telecommunication*, vol. 18, no. 5, pp. 108--113, 2011.
29. E. Mohammad and N. S. Dan, "Experimentation study of the dynamic based feedback linearization of an autonomous wheeled ground vehicle," *Robotics and Autonomous Systems*, vol. 47, pp. 47-63, 2004.
30. G. Mekonnen, S. Kumar and P. M. Pathak, "Wireless hybrid visual servoing of omnidirectional wheeled mobile robots," *Robotics and Autonomous Systems*, vol. 75, pp. 450-462, 2016.
31. C. Grand, F. Benamar and F. Plumet, "Motion Kinematics analysis of wheeled-legged robot rover 3D surface with posture adaptation," *Mechanism and machine,* vol. 45, pp. 477-495, 2010.

32. N. Chakraborty and A. Ghoshal, "Kinematics of wheeled mobile robot on uneven terrain," *Mechanism and Machine Theory*, vol. 39, pp. 1273-1287, 2004.
33. K. J. Krzysztof and M. Michal, "Optimal control of 2-wheeled mobile robot at energy performance index," *Mechanical Systems and Signal Processing*, vol. 70-74, pp. 373-386, 2016.
34. Lavalle, M. Steven, "Planning algorithm," *Cambridge University Press*, *New York,* 2006.
35. Choset, Howie, Lynch, M. Kevin and Seth, "Principle of robot motion," *Theory, algorithm and implementation. MIT Press*, 2005.
36. Seda, Milos, "Roadmap methods vs. cell decomposition in robot motion planning," *In Proceeding of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation. World Scientific and Engineering Academy and Society (WSEAS)*, pp. 127-132, 2007.
37. W. Regli, "Robot Lab: Robot path planning," *Lecture notes of department of computer science, Drexel University*, Oct 2007.
38. T. J. Schwartz and M. Sharir, "On the "Piano Movers" Problem I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers," *Communications on Pure and Applied Mathematics*, vol. 36, no. 3, pp. 345-398, 1983.
39. M. Weigl, B. Siemiaatkkowska, A. K. Sikorski and A. Borkowski, "Grid-based mapping for autonomous mobile robot," *Robotics and Autonomous Systems*, vol. 11, no. 1, pp. 13-21, 1993.
40. D. Zhu and J. C. Latombe, "New heuristic algorithms for efficient hierarchical path planning," *IEEE Transaction on Robotics and Automation*, vol. 7, no. 1, pp. 9-20, 1991.
41. G. Conte and R. Zulli, "Hierarchical path planning in a multi-robot environment with a simple navigation function," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 4, pp. 651-654, 1995.
42. H. Samet, "An overview of quadtree," *octrees and related hierarchical data structures. NATO ASI Series*, F40, 1988.
43. H. Noborio, T. Naniwa, and S. Arimoto, "A quadtree-based path-planning algorithm for a mobile robot," *Journal of Robotic Systems*, vol. 7, no. 4, pp. 555-574, 1990.
44. Choset, Howei, Burdick, Joel, "Sensor-based exploration: The hierarchical generalized Voronoi graph," *The International Journal of Robotics Research*, vol. 19, no. 2, pp. 96-125, 2000.
45. L. Lulu, A. Elnagar, "A comparative study between visibility-based roadmap path planning algorithms," *In : 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3263-3268, 2005.
46. M. D. Berg, M. V. Kreveld, M. M. Overmars and O. C. Schwarzkopf, "Computational geometry," *Springer Berlin Heidelberg*, vol. 1-17, 2000.
47. H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavriki and S. Thrun, "Principles of Robot Motion," *MIT Press, Cambridge*, MA, 2005.
48. O. Takahashi and R. J. Schilling, "Motion planning in a plane using generalized Voronoi diagrams," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 143-150, 1989.
49. C. O. Dunlaing and C. K. Yap, "A retraction method for planning the motion of a disc," *Journal of Algorithm*, vol. 6, pp. 104-111, 1985.

50. S. Garrido, L. Moreno, D. Blanco and P. Jurewicz, "Path planning for mobile robot navigation using Voronoi diagram and fast, marching," *International Journal of Robotics and Automation*, vol. 2, no. 1, pp. 42-64. 2011.
51. E. Masehian and A. Naseri, "Mobile robot online motion planning using generalized Voronoi graphs," *Journal of Industrial Engineering*, vol. 5, pp. 1-5, 2010.
52. E. Shkolink and R. Tedrake, "Path planing in 1000+dimension using a task-space Voronoi bias," *In IEEE International Conference on Robotics and Automation*, 2009.
53. P. Bhattacharya and M. L. Gavrilova, "Roadmap-based path planning-using the Voronoi diagram for a clearance for a clearance-based shortest path," *IEEE Robotics and Automation Magazine*, vol. 15, no. 2, pp. 58-66, 2008.
54. E. Masehian and M. R. Amin-Naseri, "A Voronoi diagram-visibility graph-potential field compound algorithm for robot path planning," *Journal of Robotic System*, vol. 21, pp. 275-300, 2004.
55. D. H. Yang and S. K. Hong, "A roadmap construction algorithm for mobile robot path planning using sleton maps," *Advanced Robotics*, vol. 21, no. 1-2, pp. 51-63, 2007.
56. R. Wein, J. P. Van Den Berg and D. Halperin, "The visibility-voronoi complex and its application," *Computational Geometry*, vol. 36, pp. 66-87, 2007.
57. L. E. Kavraki, P. Svestka, J. C. Latombe and M. H. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Transaction on Robotics and Automation*, vol. 12, pp. 566-580, 1996.
58. G. Sanchez and J. Latombe, "A Single-query Bidirectional Probabilistic Roadmap Planner with Lazy Collision Checking," *Springer Tracts in Advanced Robotics*, vol. 6, pp. 403-417, 2001.
59. O. Khatib, "Real time Obstacle Avoidance for manipulators and Mobile Robots," *IEEE International Conference on Robotics and Automation, Missouri*, vol. 25-28, pp. 500-505, Mar 1985.
60. G. Garibotto and S. Masciangelo, "Path planning using the potential field approach for navigation," *Fifth International Conference on Advanced Robotics*, Pisa, Italy, vol. 19-22, pp. 1679-1682, June 1991.
61. J. O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338-349, 1992.
62. J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 5, pp. 1179-1187, 1989.
63. S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots*, vol. 13, no. 3, pp. 207-222, 2002.
64. K. P. Valavanis, T. Hebert, R. Kolluru and N. Tsourveloudis, "Mobile robot navigation in 2D dynamic environments using an electrostatic potential field," *IEEE Transactions on Systems Man and Cybernetics, Part A: Systems and Humans*, vol. 30, no. 2, pp. 187-196, 2000.
65. L. Huang, "Velocity planning for a mobile robot to track a moving target-a potential field approach," *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 55-63, 2009.
66. P. Shi and Y. Zhao, "An efficient path planning algorithm for mobile robot using improved potential field," *IEEE International Conference on Robotics and Biomimetics, Guilin, China*, December, vol. 19-23, pp. 1704--1708, 2009.

67. J. Sfeir, M. Saad and H. Saliah-Hassane, "An improved artificial potential field approaches to real-time mobile robot path planning in an unknown environment," *IEEE International Symposium on Robotic and Sensors Environments, Montreal, Canada*, September vol. 17-18, pp. 208-213, 2011.
68. K. Biswas and I. Kar, "On reduction of oscillations in target tracking by artificial potential field method," *9th IEEE International Conference on Industrial and Information Systems Gwalior, India*, December vol. 15-17, pp. 1-6, 2014.
69. N. B. Hui, "Coordinated motion planning of multiple mobile robots using potential field method," *IEEE International Conference on Industrial Electronics, Control and Robotics, Orissa*, vol. 27-29, pp. 6-11, Dec 2010.
70. L. C. Pimenta, A. R. Fonseca, G. A. Pereira, R. C. Mesquita, E. J. Silva, W. M. Caminhas and M. F. Campos, "Robot navigation based on electrostatic field computation," *IEEE Transactions on Magnetics*, vol. 42, no. 4, pp. 1459-1462, 2006.
71. S. K. Pradhan, D. R. Parhi, A. K. Panda and R. K. Behera, "Potential field method to navigate several mobile robots," *Applied Intelligence*, vol. 25, no. 3, pp. 321-333, 2006.
72. D. R. Parhi and A. K. Jha, "Review and analysis of different methodologies used in mobile robot navigation," *IJAAIES*, vol. 4, no. 1, pp. 1-18, 2012.
73. D. R. Parhi and R. K. Sonkar, "Different methodologies of a navigation of autonomous mobile robot for unknown environment," *IJAICR*, vol. 4, no. 2, pp. 79-85, 2012.
74. H. J. Bremermann, "The evolution of intelligence. The Nervous system as a model of its environment," *Technical Report no.1, contract no. 477(17), Dept. Mathematics, Univ. Washington, Seattle*, July 1958.
75. J. H. Holland, "Adaptation in natural and artificila systems. Ann Aebor," *MI: University of Michigan Press*.
76. T. Shibata and T. Fukuda, "Robot motion planning by geneti algorithm with fuzzy critic," *Proc. 8th IEEE International Symp. On Intelligent Control*, pp. 565-570, Aug 1993.
77. M. T. Shing and G. B. Parkar, "Genetic algorithm for the development of real-time multi-heuristic search strategies," *Proc. 5th Conf. on Genetic Algorithm (Morgan Kaumann Publication, Los Aitos, California)*, pp. 565-570, 1993.
78. J. Xia, Z. Michalewicz, L. Zhang and K. Trojanowski, "Adaptive evolutionary planner/navigator for mobile robot," *IEEE Transcation on Evolutionary Computation*, vol. 1, no. 1, Apr 1997.
79. X. Kang, Y. Yue, D. Li and C. Maple, "Genetic algorithm based solution to dead problems in robot navigation," *International Journal of Computer application in technology*, vol. 41(¾), pp. 177-184, 2011.
80. P. Shi and Y. Cui, "Dynamic path planning for mobile robot based on genetic algorithm in unknown environment," *Proceedings of the ChineseControl and decision Conference,Xuzhou, China*, pp. 4325-4329, May 2010.
81. D. K. Pratihar, K. Deb and A. Ghosh, "Fuzzy-Genetic algorithm and time-optimal obstacle free path generation for mobile robots," *Engineering Optimization*, 32(1), 117--142, 1999.
82. N. B. Hui and D. K. Pratihar, "A comparative study on some navigation schemes of a real robot tackling moving obstacles," *Robotics and Computer-Integrated Manufacturing*, vol. 25, pp. 810-828, 2009.

83. X. Wang, Y. Shi, D. Ding and X. Gu, "Double global optimum genetic algorithm-particle swarm optimization-based welding robot path planning," *Engineering Optimization*, 2015.
84. R. Kala, "Coordination in navigation of multiple mobile robots," *Cybernetics and systems*, vol. 45, no. 1, pp. 1-24, 2014.
85. F. Liu, S. Liang and X. Xian, "Optimal robot path planning for multiple goals visiting based on Tailored genetic algorithm," *International Journal of Computational Intelligence Systems*, vol. 7, no. 6, pp. 1109-1122, 2014.
86. T. Kuyucu, I. Tanev and K. Shumohara, "Incremental evolution of fast moving and simulated snake-like robot with multi-objective GP and strongly-typed crossover," *Memetic Computing,* vol. 4, pp. 183-200, 2012.
87. P. Shi and Y. Cui, "Dynamic path planning for mobile robot based on genetic algorithm in unknown environment," *in : 2010 Chinese Control and Decision Conference, Xuzhou, China*, pp. 4325-4329, 2010.
88. S. X. Yang, Y. Hu and Meng, "A knowledge based GA path planning of multiple mobile robots indynamic environments," *in 2006 IEEE Conference on Robotics, Automation and Mechatronics, Bangkok, Thailand*, pp. 1-6, 2006.
89. Qu Hong, Ke Xing and T. Alexander, "An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots," *Neurocomputing*, vol. 120, pp. 509-517, 2013.
90. G. Carlos, S. Antonia and X. H. Vazquez, "Solving the dynamic traveling salesman problem using a genetic algorithm with trajectory prediction," *An application to fish aggregating devices. Computers and Operations Research*, vol. 56, pp. 22-32, 2015.
91. A. H. Karami and M. Hasanzadeh, "An adaptive genetic algorithm for robot motion planning in 2D complex environment," *Computers and electrical engineering*, vol. 43, pp. 317-329, 2015.
92. L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338-353, 1965.
93. Y. Wu, J. Yang and Ke Liu, "Obstacle detection and environment modeling based multisensory fusion for robot navigation," *Artificial Intelligence in Engineering*, vol. 10, no. 4, pp. 323-333, 1996.
94. D. Herrero-perez, H. Martinez-Barbara, K. LeBlanc and A. Saffiotti, "Fuzzy uncertainty modeling for grid based localization of mobile robots," *International Journal of Approximate Reasoning*, vol. 51, pp. 912-932, 2010.
95. F. Abdessemed, K. Benmahammed and E. Monacelli, "A Fuzzy-based reactive controller for a nonholonomic mobile robot," *Robotics and Autonomous Systems*, vol. 47, no. 1, pp. 31-46, 2004.
96. M. B. Montaner and A. Ramirez-serrano, "Fuzzy knowledge-based controller design for autonomous robot navigation," *Expert System with application*, vol. 14, no. 1, pp. 179-186, 1998.
97. P. G. Zavlangas and S. G. Tzafestas, "Motion control for mobile robot obstacle avoidance and navigation: a fuzzy logic-based approach," *Systems Analysis Modeling Simulation*, vol. 43, no. 12, pp. 1625-1637, 2003.
98. G. Castellano, G. Attolico and A. Distante, "Automatic generation of fuzzy rules for reactive robot controllers," *Robotics and Autonomous Systems*, vol. 22, no. 2, pp. 133-149, 1997.
99. A. Homaifar and E. McCormick, "Simultaneous design of membership function and rule sets for fuzzy controllers using genetic algorithm," *IEEE Transaction on Fuzzy systems*, vol. 3, no. 2, pp. 129-138, 1995.

100. H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcement," *IEEE transaction on neural network*, vol. 3, no. 5, pp. 724-740, 1992.
101. R. Abiyev, D. Ibrahim and B. Erin, "Navigation of mobile robot in presence of obstacles," *Advances in Engineering Software*, vol. 41, no. 10, pp. 1179-1186, 2010.
102. S. S. Ge, X. Lai and A. Mamun, "Sensor-based path planning for nonholonomic mobile robots subject to Dynamic environment," *Robotics and Autonomous System*, vol. 55, no. 7, pp. 513-526, 2007.
103. O. Motlagh, T. S. Hong and N. Ismail, "Development of new minimum avoidance systems for a behavior-based mobile robot," *Fuzzy Sets and Systems*, vol. 160, pp. 1929-1946, 2009.
104. K. Park and N. Zhang, "Dual fuzzy logic controllers for autonomous robot navigation. International Journal of Systems," *Control and Communication*, 201, vol. 3, no. 3, pp. 322-329, 2011.
105. R. Huq, G. Mann and R. Gosine, "Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation," *Applied soft computing*, vol. 8, pp. 422-436, 2008.
106. G. P. Moustris and S. G. Tzafestas, "Switching fuzzy tracking control for mobile robots under curvature constraints," *Control Engineering Practice*, vol. 19, pp. 45-53, 2011.
107. J. Gasos and A. Rosetti, "Uncertainty representation for mobile robots: Perception, modeling and navigation in unknown environments," *Fuzzy sets and systems*, vol. 107, pp. 1-24. 1999.
108. K. Demirli and I. B.Turksen, "Sonar based mobile robot localization by using fuzzy triangulation," *Robotics and autonomous systems*, vol. 33, pp. 109-123, 2000.
109. R. Carelli and E. O. Freire, "Corridor navigation and wall-following stable control for sonar-based mobile robots," *Robotics and Autonomous Systems*, vol. 45, pp. 235-247, 2003.
110. X. Rulong, W. Qiang, S. Lei and C. Lei, "Design of multi-robot path planning system based on the hierarchical fuzzy control," *Procedia Engineering*, vol. 15, pp. 235-239, 2011.
111. P. Parasuraman, "Sensor fusion for mobile robot navigation: Fuzzy associative memory," *Procedia Engineering*, vol. 41, pp. 251-256, 2012.
112. C. Flanagan, D. Toal and M. Leyden, "Subsumption and fuzzy logic, experiments in behavior based control of mobile robot," *International Journal of smart engineering system design*, vol. 5, no. 3, pp. 161-175, 2003.
113. W. Yaonan, Y. Yimin, Y. Xiaofang Z. Yi, Z. Yuanli, Y. Feng and T. Lei, "Autonomous mobile robot navigation system designed in dynamic environment based on transferable belief model," *Measurement*, vol. 44, pp. 1389-1405, 2011.
114. M. A. K. Jaradat, M. Al-Rousan and L. Quadan, "Reinforcement based mobile robot navigation in dynamic environment," *Robotics and Computer-Integrated Manufacturing*, vol. 27, pp. 135-149, 2011.
115. N. Tschichold-Gurman, "The neural network model Rule-Net and its application to mobile robot navigation," *Fuzzy Sets and System*, vol. 85, pp. 287-303, 1997.
116. A. Homaifar and E. McCormick, "Simultaneous design of membership function and rule sets for fuzzy controllers using genetic algorithm," *IEEE Transaction on Fuzzy systems*, vol. 3, no. 2, pp. 129-138, 1995.

117. M. Jaradat, M. Garibeh and E. A. Feilat, "Autonomous mobile robot planning using hybrid fuzzy potential field," *Soft Computing*, vol. 16, pp. 153-164, 2012.
118. M. Khatib and J. Saade, "An efficient data-driven fuzzy approach to the motion planning problem of a mobile robot," *Fuzzy sets and systems*, vol. 134, pp. 65-82, 2003.
119. H. Lee, J. Jung , K. Choi, P. Jiyoung and H. Myung, "Fuzzy-logic-assisted interacting multiple model (FLAIMM) for mobile robot localisation," *Robotics and Autonomous Systems*, vol. 60, pp. 1592-1606, 2012.
120. M. Hoy, A. S. Matveev and A. V. Savkin, "Collision free cooperative navigation of multiple wheeled robots in unknown cluttered environments," *Robotics and Autonomous Systems,* vol. 60, pp. 1253-1266, 2012.
121. Tae-Koo Kang, H. Zhang, Gwi-Tae Park and D. W. Kim, "Ego-motion-compensated object recognition using type-2 fuzzy set for a moving robot," *Neurocomputing*, vol. 120, pp. 130-140, 2013.
122. K. Al-Mutib, E. Mattar and M. Alsulaiman, "Implementation of fuzzy decision based mobile robot navigation using sereo vision," *Procedia Computer Sciences*, vol. 62, pp. 143-150, 2015.
123. D. N. M. Abadi and M. H. Khooban, "Design of optimal Mamdani-type fuzzy controller for nonholonomic wheeled mobile robots," *Journal of King Saud– Engineering Sciences*, vol. 27, pp. 92-100, 2015.
124. O. Castillo, H. Neyoy, J. Soria and F. Valdez, "A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot," *Applied soft computing*, vol. 28, pp. 150-159, 2015.
125. S. F. Toloue, M. R. Akbarzadeh, A. Akbarzadeh and M. Jalaeian-F, "Position tracking of a 3-PSP parallel robot using dynamic growing interval type 2-fuzzy neural control," *Applied Soft Computing*, vol. 37, pp. 1-14, 2015.
126. R. Al-Jarrah, A. Shahzad and H. Roth, "Path planning and motion coordination for multi-robots systems using probabilistic neuro fuzzy," *IFAC-papers on line*, vol. 48, no. 10, pp. 046--051. 2015.
127. F. Yili, L. Han, Z. Jiang and S. Wang, "Double layers fuzzy logic based mobile robot path planning in unknown environment," *Intelligence Automation and Soft Computing*, vol. 15, no. 2, pp. 275-288, 2009.
128. X. S. Yang, "Nature-inspired metaheuristic algorithm," *Luniver press*, 2008.
129. A. L. Christensen, R. O'Grady and M. Dorigo, "Synchronization and fault detection in autonomous robots," *Intelligent Robots and Systems*, pp. 4139-4140, 2008.
130. A. Theofanis and A. Vlachos, "Application of the firefly Algorithm for solving the economic emission load dispatch problem," *International Journal of Combinatorics*, 2011.
131. S. Leandro dos, A. Diego Luis and M. Viviana, "A chaotic firefly algorithm applied to reliability-redundancy optimization," *IEEE conference*, pp. 517-521, 2011.
132. A. H. Gandomi, X. S. Yang and A. H. Alavi, "Mixed variable optimization using firefly algorithm, *Computers and Structures*, vol. 89, pp. 2325-2336, 2011.
133. R. Falcon, X. Li, A. Nayak and I. Stojmenovic, "A Harmony-seeking firefly swarm to the periodic replacement of damaged sensors by a team of mobile robots, *IEEE International Conference*, pp. 4914-4918, 2012.

134. A. Rahmani and S. A. MirHassani, "A hybrid firefly-genetic algorithm for the capacitated facility location problem," *Information Sciences*, vol. 283, pp. 70-78, 2014.
135. M. Mitic and Z. Mijkovic, "Bio inspired approach to learning robot motion trajectories and visual control commands," *Expert Systems with Application*, vol. 42, no. 5, pp. 2624-2637, 2015.
136. A. G. Roy, P. Rakshit, A. Konar, S. Bhattacharya, E. Kim and A. Nagar, "Adaptive firefly algorithm for nonholonomic motion planning of car like system," *IEEE Congress on Evolutionary Computation*, pp. 2162-2169, 2013.
137. N. Babak and M. R. Meybodi, "Speciation based firefly algorithm for optimization in dynamic environment," *International Journal of Artificial Intelligence*, vol. 8, no. 12, pp. 118-132, 2012.
138. M. Brand and X. H. Yu, "Autonomous robot path optimization using firefly algorithm," *Proceedings of the 2013, International Conference on Machine Learning and Cybernetics, Tianjin*, pp. 14-17, 2013.
139. M. Farahami, A. A. Abshouri, B. Nasiri and M. R. Meybodi, "A Guassian firefly algorithm," *International Journal of Machine Learning and Computing*, vol. 1, no. 5, pp. 448, 2011.
140. S. L. Tilahun and H. C. Ong, "Modified firefly algorithm," *Journal of Applied Mathematics*, 2012.
141. A. Baykasoglu and F. B. Ozsoydan, "Adaptive firefly algorithm with chaos for mechanical design optimization problems," *Applied Soft Computing*, vol. 36, pp. 152-164, 2015.
142. M. Li, Y. Zhang, B. Zeng, H. Zhou and J. Liu, "The modified firefly algorithm considering fireflies visual range planning and its application in assembly sequences planning," *International Journal of Advance Manufacturing Technology*, vol. 82, no. 5-8, pp. 1381-1403, 2016.
143. S. Sivakumar and R. Venkatesan, "Meta-heuristic approaches for minimizing error in localization of wireless sensor networks," *Applied Soft Computing*, vol. 36, pp. 506-518, 2015.
144. X. S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Engineering with Computers*, vol. 29, pp. 175-184, 2013.
145. Baykasoglu and B. O. Fehmi, "An improved firefly algorithm for solving dynamic multidimensional knapsack problems," *Expert systems with application*, vol. 41, pp. 3712-3725, 2014.
146. M. Alweshah and S. Abdullah, "Hybridizing firefly algorithm with a probabilistic neural network for solving classification problems," *Applied Soft Computing*, vol. 35, pp. 513-524, 2015.
147. M. K. Kaushik and V. Aroro, "A Hybrid data clustering using firefly algorithm based improved genetic algorithm," *Procedia Computer Science*, vol. 58, pp. 249-256, 2015.
148. D. Zouache, F. Nouioua and A. Moussaoui, "Quantum-inspired firefly algorithm with particle swarm optimization for discrete optimization problems," *Soft Computing*, pp. 1-19, 2015.
149. A. Baykasoglu and F. B. Ozsoydan, "Adaptive firefly algorithm with chaos for mechanical design optimisation problems," *Applied soft computing*, vol. 36, pp. 152-164, 2015.
150. A. Hidalgo-Paniagua, A. Miguel, Vega–Rodriguez, J. Ferruz and N. Pavon, "Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach," *Soft Computing*, vol. 1-16, 2015.

151. M. Brand and Xiao-Hua Yu, "Autonomous robot path optimisation using firefly algorithm," *International Conference on Machine Learning and Cybernetics, Tianjin*, vol. 3, pp. 14-17, 2013.
152. D. Sutantyo and P. Levi, "Decentralized underwater multi robot communication using bio-inspired approaches," *Artificial Life Robotics*, vol. 20, pp. 152-158, 2015.
153. D. Sutantyo, P. Levi, C. Moslinger and M. Read, "Collective-Adaptive Leavy Flight for Underwater multi-robot exploration," *International conference on mechatronics and automation*, pp. 456-462, 2013.
154. A. L. Christensen, G. Rehan O' and Dorigo, "Synchronisation and fault detection in autonomous robots," *IEEE/ RSJ Intelligent Conference on Robots and Systems*, pp. 4139-4140, 2008.
155. G. Wang, L. Guo, D. Hong, H. Duan, L. Liu and H. Wang, "A modified firefly algorithm for UCAV path planning," *International Journal of Hybrid Information Technology*, vol. 5, no. 3, pp. 123--144, July 2012.
156. D. Janglova, "Neural networks in mobile robot motion," *International Journal of Advanced Robotic Systems*, vol. 1, pp. 15-22, 2004.
157. B. Siemiatkowska and A. Dubrawski, "Cellular neural networks for navigation of a mobile robot," *Rough Sets and Current Trends in Computing*, vol. 1424, pp. 147-154, 1999.
158. X. Wang, Zeng-Guang Hou, A. Zou, M. Tan and C. Long, "A behaviour conroller based on spiking neural networks for mobile robots," *Neurocomputing*, vol. 71, no. 4-6, pp. 655-666, 2008.
159. J. Qiao, R. Fan, H. Han and X. Ruan, "Q-learning based on dynamical structures neural network for robot navigation in unknown environment," *Advances in neural network*, vol. 5553, pp. 188-196, 2009.
160. Q. L. Li, Y. Song and Z. G. Hou, "Neural network based Fast SLAM for automobile robots in unknown environments," *Neurocomputing*, vol. 165, pp. 99-110, 2015.
161. Y. K. Na and S. Y. Oh, "Hybrid control for autonomous mobile robot navigation using neural network based behavior modules and environment classification," Autonomous robots, vol. 15, pp. 193-206, 2003.
162. J. K. Pothal and D. R. Parhi, "Navigation of multiple robots in a highly clutter terrains using adaptive neuro-fuzzy inference system," *Robotics and Automation*, vol. 72, pp. 48-58, 2015.
163. A. Abu Baker, "A novel mobile robot navigation system using neuro-fuzzy rule - based optimization technique," *Res. J. Appl. Sci. Eng. Technol.*, vol. 4, no. 15, pp. 2577-2583, 2012.
164. P. K. Pal and A. Kar, "Sonar-based mobile robot navigation through supervised learning on a neural net," *Autonomous Robots*, vol. 3, pp. 355-734, 1996.
165. A. Medina-Santiago, J. L. Campus-Anzueto, J. A. Vazquez-Feijoo, H. R. Hernandez-de-Leon and R. Mota-Grajales, "Neural control systems in obstacle avoidance in mobile robots using untrasonic sensors," *Journal of Applied Research and Technology*, vol. 2, pp. 104-110, 2014.
166. G. Capi, S. Kaneko and B. Hua, "Neural network based guide robot navigation: An evolutionary approach," *Procedia Computer Science*, vol. 76, pp. 74-79, 2015.
167. U. A. Syed, F. Kunwar and M. Iqbal, "Guided autowave pulse coupled neural network (GAPCNN) based real time path planning and an obstacle avoidance

scheme for mobile robots," *Robotics and Autonomous Systems*, vol. 62, pp. 474-486, 2014.
168. Z. Hendzel, "Neural network reacive navigation and control of wheeled mobile robot," *Neural Networks and Soft Computing*, vol. 19, pp. 686-691, 2003.
169. B. Markoski, S. Vukosavavijev, D. Kukolj and S. Pletl, "Mobile robot control using self-learning neural network," *Intelligent systems and informatics*, pp. 45-48, 2009.
170. Y. Quinonez, M. Ramirez, C. Lizarraga, I. Tostado and J. Bekios, "Autonomous robot navigation based on pattern recognition techniques and artificial neural networks," *Advanced Intelligent Computing*, vol. 6838, pp. 210-217, 2015.
171. R. C. Eberhart and J. A. Kennedy, "A new optimizer using particle swarm theory. Proceedings of the sixth international symposium on micro machine and human science," *IEEE service centre, Piscataway, NJ, NAGOYA, Japan*, pp. 39-43, 1995.
172. X. Tang, L. Li and B. Jiang, "Mobile robot SLAM method based on multi-agent particle swarm optimized particle filter," *The Journal of China Universities of Posts and Telecommunications*, vol. 21, no. 6, pp. 78-86, 2014.
173. V. H. Xuan, H. Cheolkeun and L. Jewon, "Novel hybrid optimisation algorithm using PSO and MADS for the trajectory estimation of a four track wheel skid-steered mobile robot," *Advanced Robotics*, vol. 27, no. 18, pp. 1421-1437, 2013.
174. A. Atyabi, S. Phon-Amnuaisuk and C. K. Ho, "Applying area extension PSO in robotic swarm," *Journal of Intelligent Robot System*, vol. 58, pp. 253-285, 2010.
175. Q. Tang and P. Eberhard, "Cooperative motion of swarm mobile robots based on particle swarm optimisation and multibody system dynamics," *Mechanics based Design of Structures and Machines*, vol. 39, no. 2, pp. 179-193, 2011.
176. M. S. Couceiro, R. P. Rocha and F. Nuno, "A PSO multi-robot exploration approach over unreliable MANETs," *Advanced Robotics*, vol. 27, no. 16, pp. 1221-1234, 2013.
177. Y. Chen, J. Cheng, C. Lin, X. Wu, Y. Ou and Y. Xu, "Classification based learning by particle swarm optimization for wall-following robot navigation," *Neurocomputing*, vol. 113, pp. 27-35, 2013.
178. P. K. Das, H. S. Behera and B. K. Panigrahi, "A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning," *Swarm and Evolutionary Computation*, 2016.
179. He Bo, L. Ying, S. Zhang, X. Feng, T. Yan, R. Nian and Y. Shen, "Autonomous navigation based on unscented-Fast SLAM using particle swarm optimization for autonomous underwater vehicles," *Measurement*, vol. 71, pp. 89-101, 2015.
180. M. Algabri, M. Hassan, R. Hedjar and M. Alsulaiman, "Comparative study of soft computing technique for mobile robot navigation in an environment," *Computers in Human Behavior*, vol. 50, pp. 42-56, 2015.
181. D. Macro, "Ant colony system: A Cooperative learning approach to the travelling salesman problem," *IEEE transaction on evolutionary computation*, vol. 1, no. 1, pp. 53-66, 1997.
182. T. Guan-Zheng, HE Huan and S. Aaron, "Ant colony system algorithm for real time globally optimal path planning of mobile robots," *Acta Automatica Sinica*, vol. 33, no. 3, pp. 279-285, 2007.
183. S. Liu, L. Mao and J. Yu, "Path planning based on ant colony algorithm and distributed local navigation for multi-robot systems," *Proceedings of the 2006 IEEE International Conference of Mechatronics and Automation*, pp. 1733-1738, 2006.

184. O. Castillo, H. Neyoy, J. Soria, P. Melin and F. Valdez, “A New approach for dynamic fuzzy logic parameter tuning in ant colony optimisation and its application in fuzzy control of a mobile robot,” *Applied soft computing*, vol. 28, pp. 150-159, 2015.
185. F. K. Purian and E. Sadeghian, “Mobile robots path planning using ant colony optimization and fuzzy logic algorithm in unknown dynamic environment. International Conference on Control, Automation,” *Robotics and Embedded systems (CARE)*, pp. 1-6, 2013.
186. P. K. Mohanty and D. R. Parhi, “Optimal path planning for a mobile robot using cuckoo search algorithm,” *Journal of Experimental and Theoretical Artificial Intelligence*, 2014.
187. M. A. Hossain and I. Ferdous, “Autonomous path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique,” *Robotics and Automation Systems*, vol. 64, pp. 137-141, 2015.
188. D. R. Parhi and A. K. Jha, “Path planning of mobile robot using bacterial foraging optimization, *IJAICR*, vol. 4, no. 1, pp. 1-5, 2012.
189. X. Liang, L. Li, J. Wu and H. Chen, “Mobile robot path planning based on adaptive bacterial foraging algorithm,” *Journal of South University*, vol. 20, no. 12, pp. 3391-3400, 2013.
190. H. H. Liang and C. Lee, “Efficient collision-free path planning of multiple mobile robots system using efficient artificial bee colony algorithm,” *Advances in Engineering Software*, vol. 79, pp. 47-56, 2015.
191. A. Marco, C. C. A. Victor and H. Uriel, “Mobile robot path planning using artificial bee colony and evolutionary programming,” *Applied Soft Computing*, vol. 30, pp. 319-328, 2015.
192. P. K. Mohanty, K. Sandeep, D. R. Parhi, “A new ecologically Inspired Algorithm for mobile robot navigation,” *Proceedings of the 3rd International Conference on frontiers of Intelligent Computing: Theory and Application*, pp. 755-762, 2014.
193. S. Kundu and D. R. Parhi, “Modified shuffled frog leaping algorithm 6 DOF motion for underwarer mobile robot,” *International Conference on Computational Intelligence: Modelling, Techniques and Applications, Procedia Technology*, vol. 1, pp. 295-303, 2013.
194. J. Ni, X. Yin, J. Chen and X. Li, “An Improved shuffled frog leaping algorithm for robot path planning,” *10th IEEE International Conference on Natural Computation*, pp. 545-549, 2014.
195. J. Guo, Yu Gua and G. Coi, “The path planning for mobile robot based on bat algorithm,” *International Journal of Automation and Control*, vol. 1, pp. 50-60, 2015.
196. N. Zhang, D. Beetner, D. Wunch, B. Hemmelman, A. Hasan, “An embedded real-time neuro-fuzzy controller for mobile robot navigation,” *IEEE international conference on fuzzy system*, pp. 319-324, 2005.
197. M. Wang and N. James, “Fuzzy logic-based real-time navigation in unknown environment with dead ends,” *Robotics and autonomous systems*, vol. 56, pp. 625--643, 2008.
198. M. A. Garcia, O. Moutiel, O. Castillo, R. Sepulveda and P. Melin, “Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation,” *Applied soft computing*, vol. 9, pp. 1102-1110, 2009.
199. M. Algabri, H. Mathkour, H. Ramdane and M. Alsulaiman, “Comparative study of soft computing technique for mobile robot navigation in an unknown environment,” *Computers in human behavior*, vol. 50, pp. 42-56, 2015.

200. M. M. Joshi and M. A. Zaveri, “Reactive navigation of autonomous mobile robot using neuro-fuzzy system,” *International Journal of Robotics and Automation*, vol. 2, no. 3, pp. 128-145, 2011.
201. J. Wang, Y. Zhang and L. Xia, “Adaptive genetic algorithm enhancement for path planning of mobile robots,” *International Conference of Measuring Technology and Mechatronics Automation (ICTMA), Changsha, China*, vol. 3, pp. 416-419, 2010.
202. L. Cherroun and M. Boumehraz, “Fuzzy behavior-based navigation approach for mobile robot in unknown environment,” *Journal of Electrical Engineering*, vol. 13, pp. 1-8, 2013.
203. Z. Su, B. Zeng, G. Liu, F. Ye and M. Xu, “Application of fuzzy neural network in parameter optimization of mobile robot path planning using potential field, *IEEE International Symposium on Industrial Electronics, Vigo, Spain*, vol. 4, no. 7, pp. 2125-2128, June 2007.
204. W. Shi, K. Wang and S. X. Yang, “A fuzzy-neural network approach to multisensor integration for obstacle avoidance of a mobile robot,” *Intelligent Automation and Soft Computing*, vol. 15, no. 2, pp. 289-301, 2009.
205. H. Mo, Q. Tang and L. Meng, “Behavior-Based Fuzzy Control for Mobile Robot Navigation,” *Mathematical Problems in Engineering*, 2013.
206. M. M. Joshi and M. A. Zaveri, “Neuro-fuzzy based autonomous mobile robot navigation system,” *IEEE International Conference on Control Automation Robotics and Vision, Singapore*, vol. 7, no. 10, pp. 384-389, Dec 2010.
207. I. Engedy and G. Horvath, “Artificial Neural Network based Local Motion Planning of a Wheeled Mobile Robot,” *In IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, pp. 213-218, 2010.

Dissemination

International journals:

1. B. K. Patle, D. R. K. Parhi, A. Jagadeesh, Sunilkumar Kashyap. Probabilistic fuzzy controller based robotics path decision theory. World Journal of Engineering, 2016.
2. B. K. Patle, D. R. K. Parhi, A. Jagadeesh, O.P. Sahu. Real time navigation approach for mobile robot. Journal of Computers, 2015.
3. D.R.K. Parhi, B.K. Patle, A. Jagadeesh. Mobile Robot Navigation Considering the different method: A review. International Journal of Artificial Intelligence and Computational Research (IJAICR), 2(3), 125--134, 2011.
4. D.R.K. Parhi, B.K. Patle, A. Jha, S. Mahapatra. Analysis of map representation for control of mobile robot in various environmental situations. International Journal of Applied Artificial Intelligence in Engineering System, 5(2), 151--162, 2013.
5. D.R.K. Parhi, B. K. Patle, S. Mahapatra. Brains in the artificial world. International Journal of Artificial Intelligence and Computational Research (IJAICR), 4(1), 33--38, 2012.
6. B.K.Patle, D.R.K. Parhi, A. Jagadeesh, Sunilkumar Kashyap. On Firefly Algorithm: Optimization and Application in Mobile Robot Navigation. World Journal of Engineering, (In Press).
7. B.K.Patle, Anish Pandey, D.R.K. Parhi, A. Jagadeesh. Robot Path Control in Uncertain Environment using Firefly Algorithm. Sadhana - Academy Proceedings in Engineering Science (Communicated).
8. B.K.Patle, D.R.K. Parhi, A. Jagadeesh, Sunilkumar Kashyap. Matrix-Binary Codes based Genetic Algorithm for Path Planning of Mobile Robot. Computers and Elactrical Engineering, (Communicated).

International Conference:

1. B. K. Patle, S. Mahapatra, A.K. Jha, D.R.K. Parhi. Cell decomposition: the way for path planning in complex environment. Proceedings of International Conference on Computing, Communicationand Application (ICCCA), Dindigul, Tamil Nadu, 138--142, Feb 2012.
2. B. K. Patle, Alok Kumar Jha, Dayal R Parhi. Design of fuzzy logic avigational Controller for autonomous mobile robot. In: Proceedings of Shastrarth an International Conference on Eco Friendly Technologies in Mechanical Engineering for Sustainable Growth, Bhilai, FEB 8-9, 2013.
3. B. K. Patle, S. Mahapatra, D.R.K. Parhi, A.K. Jha,. Evolutionary algorithms and related techniques in the field of robotics: A review. Proceedings of International Conference on Advances in Modelling, Optimization and Computation (AMOC), Roorkee, 36-37, Dec 2011.
4. B. K. Patle, S. Mahapatra, D.R.K. Parhi. Navigation of a mobile bobot using fuzzy logic. Proceedings of International Conference on Computer Science Engineering (ICCSE), Nagpur, Maharashtra, 128-130, Feb 2012.

5. S. Mahapatra, B. K. Patle, A.K. Jha, D.R.K. Parhi. Fuzzy Logic Control of a WMR. In: Proceedings of International Conference on Computing, Communication and Applications (ICCCA), Dindigul, Tamilnadu, 1-5, Feb 2011.
6. Alok Kumar Jha, B K Patle, Dayal R Parhi, "Design of a Behavior based Navigational Controller using Fuzzy Logic. In: Proceedings of Shastrarth an International Conference on Eco Friendly Technologies in Mechanical Engineering for Sustainable Growth, Bhilai, FEB 8-9, 2013.

Bio-Data of the Candidate

Name of the Candidate : Bhumeshwar Patle
Father's Name : Kunjilal Patle
Permanent Address : Ayodhya Nagar, Lohiya Ward, Ring Road, Gondia
Dist.- Gondia, Maharastra
Email : balu_patle@rediffmail.com

Academic Qualification

Examination	Board/University	Year of passing	Division	Subject
Ph.D.	Pursuing at NIT Rourkela	-	-	Mechanical Engineering
M.Tech	P.R.M.I.T.R. Badnera	2009	1st	Mechanical Engineering
B.E.	G.E.C. Chandrapur	2007	1st	Mechanical Engineering
HSSC	Pune State Board Maharastra	2002	1st	Science
HSC	Pune State Board Maharastra	2000	1st	All